Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

2021-09

MULTI-ARMED BANDIT MODELS FOR EXPLOITATION OF CYBER NETWORKS

Chan, Baixian Alvin

Monterey, CA; Naval Postgraduate School

http://hdl.handle.net/10945/68308

Copyright is reserved by the copyright owner.

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MULTI-ARMED BANDIT MODELS FOR EXPLOITATION OF CYBER NETWORKS

by

Baixian Alvin Chan

September 2021

Thesis Advisor:	Roberto Szechtman
Co-Advisor:	James Grant, Lancaster University
Second Reader:	Moshe Kress

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2021	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE MULTI-ARMED BANDIT MODELS FOR EXPLOITATION OF CYBER NETWORKS	5. FUNDING NUMBERS
6. AUTHOR(S) Baixian Alvin Chan	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

Computer networks are often the target of cyber attacks carried out by malevolent agents, to either disable critical system operations or to surreptitiously gain access to sensitive data. The asymmetric and covert nature of cyber attacks has led to their increased prevalence, where high-impact attacks on critical infrastructure can be launched with minimal resources. We consider the setup of a network switch and its connected nodes, and use multi-armed bandit models as a framework to formulate a network attack strategy to maximize expected rewards earned over time. Such models present upper confidence bound–based approaches on the cumulative regret, through optimal choice of possible attacker actions over a finite time horizon and bounded action space. We evaluate relevant multi-armed bandit models and develop our own algorithm. Numerical simulations consistently suggest that low cumulative regret is achieved over time for our algorithm in comparison to the other algorithms evaluated. We thus present a stylized model for strategic network exploitation, with the attacker having no prior knowledge of the rewards of various nodes in a network with a star topology. This enables effective network defenses to be continually developed, based on specific network topologies and reward feedback mechanisms.

14. SUBJECT TERMS multi-armed bandit models, exploitation, cyber networks	15. NUMBER OF PAGES 91
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

MULTI-ARMED BANDIT MODELS FOR EXPLOITATION OF CYBER NETWORKS

Baixian Alvin Chan
Civilian, Singapore Technologies Electronics
B.Eng (Electrical and Electronic Engineering), Nanyang Technological University, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: Roberto Szechtman
Advisor

James Grant
Co-Advisor

Moshe Kress
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Computer networks are often the target of cyber attacks carried out by malevolent agents, to either disable critical system operations or to surreptitiously gain access to sensitive data. The asymmetric and covert nature of cyber attacks has led to their increased prevalence, where high-impact attacks on critical infrastructure can be launched with minimal resources. We consider the setup of a network switch and its connected nodes, and use multi-armed bandit models as a framework to formulate a network attack strategy to maximize expected rewards earned over time. Such models present upper confidence bound–based approaches on the cumulative regret, through optimal choice of possible attacker actions over a finite time horizon and bounded action space. We evaluate relevant multi-armed bandit models and develop our own algorithm. Numerical simulations consistently suggest that low cumulative regret is achieved over time for our algorithm in comparison to the other algorithms evaluated. We thus present a stylized model for strategic network exploitation, with the attacker having no prior knowledge of the rewards of various nodes in a network with a star topology. This enables effective network defenses to be continually developed, based on specific network topologies and reward feedback mechanisms.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Modeling Approach 2

1.3 Research Questions 3

1.4 Thesis Outline 4

2 Literature Review **5**

2.1 Classical Multi-armed Bandits Problem 5

2.2 Review on Multi-armed Bandit Models 8

2.3 Review on Multi-armed Bandits in Cyber Models 12

3 Problem Formulation and Methodology **15**

3.1 Background 16

3.2 The Global Confidence Bound Learning Algorithm 18

3.3 LiEtAl Algorithm for Stochastic Online Learning 27

3.4 Our Proposed Algorithm 34

3.5 Summary 39

4 Results and Analysis **41**

4.1 Global Confidence Bound Learning Algorithm 41

4.2 LiEtAl Algorithm 46

4.3 Our Proposed Algorithm 57

5 Conclusion **65**

List of References **67**

Initial Distribution List **69**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 3.1 Typical Local Area Network Topology, with $\mathcal{K}$ Nodes. 15

Figure 4.1 Cumulative Regret of the GCB Algorithm versus Time. 44

Figure 4.2 GCB Algorithm - Cumulative Regret Curves. 45

Figure 4.3 Plot of Chosen Arm - Single Sample Path. 47

Figure 4.4 Plot of Expected Cumulative Regret versus Time. 48

Figure 4.5 Plot of Chosen Arm versus Time - Single Sample Path. 50

Figure 4.6 Plot of Expected Cumulative Regret versus Time. 50

Figure 4.7 LiEtAl Algorithm - Cumulative Regret Curves. 51

Figure 4.8 Plot 1 of Cumulative Regret Curves for Various β's. 53

Figure 4.9 Plot 2 of Cumulative Regret Curves for Various β's. 54

Figure 4.10 Varying Constant Value: Plot of Cumulative Regret versus Time. 56

Figure 4.11 Proposed Algorithm - Cumulative Regret Curves. 57

Figure 4.12 Cumulative Regret Curves (with Super-arm) for Various β versus η. 59

Figure 4.13 Cumulative Regret Heatmap for Parameter Values of η versus β . 60

Figure 4.14 Plot of $\beta(\eta)$ versus η. 61

Figure 4.15 Selected Cumulative Regret Curves for η without Super-arm. . . 62

Figure 4.16 Cumulative Regret Bounds for Various η, without Super-arm. . . 63

Figure 4.17 Averaged Cumulative Regret Curves for Various Algorithms. . . 64

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 4.1 True Means for Both Cases. 41

Table 4.2 LiEtAl Algorithm - Summary Table 1 47

Table 4.3 LiEtAl Algorithm - Summary Table 2 49

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

GCB	Global Confidence Bound
IDS	Intrusion Detection System
KL	Kullback–Leibler
LAN	Local Area Network
MAB	Multi-armed Bandit
NPS	Naval Postgraduate School
UCB1	Upper Confidence Bound 1

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Cyberspace can be broadly considered as a complex digital environment formed when people and software interact virtually over the Internet via computer networks and information technology infrastructures. However, cyber networks have been subjected to sophisticated and targeted threats on an increasingly prevalent basis. The aim of such attacks would be to either disrupt key systems' operations or to surreptitiously gain access to critically sensitive data exchanged over these cyber networks. The scale of potential attacks on cyber networks can be massive, leading to disruptions to people's lives. Thus, this research deals with the modelling of attacks against cyber networks.

Network defense mechanisms can be made more robust by using simulated 'attacks' to identify potential weaknesses in the network, and then improving defense mechanisms to handle such identified network vulnerabilities. This work serves to provide a framework for a potential simulated adversary to plan his operation to strategically exploit a given target network and maximize expected reward, without prior knowledge of the target network topology and the expected reward associated within each elemental node of the network. A coordinated cyber attack (CCA) vector can be broadly defined as the attack path containing the sequence of possible action choices that a potential attacker executes, in order to attack the network. In [1], it is mentioned that multi-armed bandit models are suitable for studying this class of problems as they can be applied into allocating limited resources between competing choices to derive CCA vectors that typically form the most vicious attack goal. The study further mentions that the CCA vector does not require the potential attackers to have prior knowledge of the target network, and also considers the behavioral stochasticity of the attackers using a Markov-game model to formulate the optimal attack strategy [1].

We consider in detail the topology of a simple local area network consisting of a single network switch and its connected nodes. We suggest a policy for an attacker, with a specific "star" network topology, that can be adopted in order to achieve an optimal balance between exploration and exploitation of the key network nodes in order to achieve maximum expected reward. We derive the theoretical guarantees for the performance of this policy and verify them with simulation. The insights obtained from this network configuration can be extended to formulating attack strategies for more complex network topologies.

We evaluate relevant state of the art multi-armed bandit models as part of our work to formulate an optimal possible strategy that the attacker should adopt to achieve maximum reward.

We first evaluate the performance of the global confidence bound algorithm presented in [2] that was developed by considering the case of a combinatorial stochastic partial monitoring game. In such a game that is conducted repeatedly between the attacker and the environment (network configuration), the attacker executes an action in every time step and the algorithm attempts to formulate an optimal attack strategy by suggesting optimal actions that the attacker should adopt to maximize expected reward and thus minimize expected regret.

We also evaluate another algorithm that was formulated for a stochastic online learning application for a sequential decision problem [3]. Since the algorithm presented in [3] is more complex in the general setting, we adapted the algorithm to our defined problem. For each action played by the attacker in each time step, the attacker obtains a sample reward feedback corresponding to that action played with corresponding independent probability. The algorithm was designed to have upper bounds on the possible regret corresponding to the asymptotic lower bounds of regret in this case.

Lastly, we propose our own improved algorithm to address our defined problem and present its derivation and associated analytic results. Results obtained from numerical simulations indicate that our algorithm achieves consistently lower cumulative regret over time, in comparison to the other algorithms evaluated in this work. Parts of the theoretical analysis relating to our proposed algorithm are left for future work.

List of References

[1] P. Guan and J. Wang, "Optimal adaptive coordinated cyber-attacks on power grids using ϵ-greedy method," in *2019 North American Power Symposium (NAPS)*, 2019 [Online]. Available: https://ieeexplore.ieee.org/document/9000385

[2] T. Lin, B. Abrahao, R. Kleinberg, J. Lui, and W. Chen, "Combinatorial partial monitoring game with linear feedback and its applications," in *Proceedings of the 31st International Conference on Machine Learning*, 2014 [Online]. Available: http://proceedings.mlr.press/v32/lind14.html

[3] S. Li, W. Chen, Z. Wen, and K.-S. Leung, "Stochastic online learning with probabilistic graph feedback," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, February 2020 [Online]. Available: https://www.microsoft.com/en-us/research/publication/stochastic-online-learning-with-probabilistic-graph-feedback/

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First and foremost, I would like to thank my family for your unwavering support throughout this journey so that I could focus on the academic commitments for this course. This was all the more challenging given the circumstances arising from the COVID-19 pandemic.

Next, I would like to sincerely thank both of my thesis advisors, Professor Roberto Szechtman and Professor James Grant, for your patient guidance throughout this time. Your knowledgeable insights on the subject matter, advice, devotion and enthusiasm definitely made this an equally memorable and exciting journey. This was all the more important given that we had to meet virtually across various different time zones during the time of the COVID-19 pandemic. I am ever grateful for all the kind patience and guidance along this incredible journey, especially on a research topic that was totally new to me. I could not have asked for more as my thesis advisors.

I would also like to extend my thanks to my second reader, Distinguished Professor Moshe Kress, for the very insightful comments and keeping me on target for completing this thesis.

Lastly, I would like to specially thank my fellow course mate and dearest best friend, Ng Wei Xiang Nicholas. I am grateful for your unwavering friendship, company and dedicated support throughout this academic journey at NPS. Thank you for always being such a true friend, and for always making the many daily ordinary moments really extraordinary. I could not have asked for more. Let us always cherish this precious, yet amazing friendship and bond.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

This chapter highlights the underlying motivations and operational significance behind our research problem. Various novel aspects of our work and contributions will be presented as well.

1.1 Background and Motivation

Cyberspace can be broadly considered to be the digital environment that is made up of numerous devices which are interconnected via electronic networks, to facilitate information exchange. It can be regarded as the global commons in which people from all corners of the globe remain interconnected for a wide range of interactions such as social and economic activities. In the technological age, cyberspace will be a critical battlespace. It is a space that is not only uncertain and dynamic, but often favoring the freedom of movement of hostile actors over the best efforts of network defenders. In [1], the authors highlighted that threats to computer networks within the cyber domain have evolved to be more sophisticated with increased prevalence, often times destructive and coercive. It is thus paramount that defense mechanisms to secure key cyber networks are robust, while cyber defenders remain constantly vigilant and be on top of the game.

Broadly, attacks on computer networks are conducted to either disable and disrupt key system operations or to surreptitiously gain access to critical data that is exchanged within the network. Our work focuses on attacks that seek to disable network operations as these can potentially result in devastating consequences, especially on the networks of critically important system infrastructure. The authors in [2] mentioned that there has been an increased proliferation of such activities by various state or non-state actors given the asymmetric and covert nature of such operations, where only minimal resources are required to launch high-impact attacks on key targets. It is a constant battle to protect computer networks against potential cyber attacks, with the continuous challenge of developing cyber defense capabilities that are effective, relevant and yet robust.

Many conventional network defense tools harness static knowledge of existing system

threats and known vulnerabilities. However, recent attacks have demonstrated that defense tools based on traditional approaches are no longer adequate as the nature of computer network attacks get increasingly sophisticated and continually evolve. Simulated network attacks offer a possibility for identifying potential gaps in network defense systems and mechanisms.

Hence, the use of multi-armed bandit models enable optimal and effective network attack strategies to be devised. Similar to the modus operandi associated with certain classes of cyber attacks, multi-armed bandit models start off with a limited understanding of the network environment. With limited knowledge of the network, the model tries to identify the most promising nodes to attack within a finite time interval to cause maximum damage. These models simulate a dimension of attack realism, and are definitely useful for formulating realistic network attack strategies.

1.2 Modeling Approach

Computer networks are comprised of numerous connected devices, resulting in networks with heterogenous topologies and thus make designing potential attack models a rather complex affair. In this research, we consider a stylized model for the attacker(s) in which the target network is strategically exploited with scant prior knowledge of the network topology. Such a model is strategic as it enables the attacker to confidently identify and conduct repeated attacks on the optimal network node within the finite time horizon, instead of concentrating attacks on the other sub-optimal network nodes. Using the concept of multi-armed bandit models, different models are explored and applied to our defined scenario in this work.

Multi-armed bandit models are applicable when a decision-maker needs to maximize the possible reward gained given the lack of complete information about the distributions and associated true reward means, via an optimal sequence of action choices from a finite action space. The models typically conduct initial explorations of the network, in attempt to identify the most promising network node(s) to exploit over time. The choice of action on whether to explore or exploit results in a regret, which arises as a result of not selecting the most optimal action in any particular time step. Multi-armed bandit models therefore seek to present optimal upper confidence bound approaches on achieving minimal regret.

In our scenario, we considered a network with a predefined number of nodes where each node in the network has a respective reward value associated with it. The reward value gives the relative importance of each node as target, and serves as useful information for the attacker to determine which nodes are important and critical for operation of the network. In each time step, the operation of a particular network node becomes disrupted when an attack is executed on that node. As a result of attacking a particular node, the attacker also gains a corresponding observation on the associated node reward value. This observation is manifested as an independent Bernoulli random variable outcome with unknown mean p, where p is unique to each node. The attacker would then be able to estimate the associated reward of each node from the respective Bernoulli observations obtained over time, and formulates the attack strategy accordingly.

The optimal attack strategy would then be one that enables the attacker to first learn about which nodes are likely to be the more promising nodes based on the respective estimated node reward values, via an initial exploration phase. The knowledge gained about the estimated rewards of the various network nodes then enables the attacker to strategically formulate an optimal set of actions during the subsequent exploitation phase, in order to maximize his objective of disabling critical network operations by attacking the most rewarding network node within the finite time interval before getting caught. The most important node in the network would therefore be the one with the largest associated reward value.

The stylized model proposed provides a suggested design framework for devising possible optimal network attack strategies, by harnessing the right balance of exploration and exploitation choice actions within a finite time interval. The insights gained from the optimal adversarial attack approaches developed using multi-armed bandit models will greatly aid in the design of robust cyber network defense systems.

1.3 Research Questions

The main questions that we aim to address in this research are:

- Within a network, with no prior knowledge about the network topology, how can the attacker exploit the network to maximize total expected reward within a finite time interval? The reward to be maximized would be the number of times the optimal node

is attacked over time, in comparison with the other sub-optimal nodes.

- What are the key parameters that affect the attacker's strategy to exploit the cyber network?
- How do the respective regret performance of the various explored algorithms vary for our defined scenario?

1.4 Thesis Outline

In this chapter, we provided an overview into the problem investigated in this work together with the underlying motivations and significance to real-world applications. We described the modeling approach that was adopted, together with the key research questions investigated as part of our work.

Chapter 2 provides a comprehensive overview of the concept of multi-armed bandits and reviews the advanced bandit algorithms explored in this work. Additionally, we review existing research on the application of multi-armed bandit models to cyber network defense problems.

In Chapter 3, we introduce our multi-armed bandit model for a cyber attack scenario, and present a theoretical analysis of the problem.

Chapter 4 presents the various results and insights obtained from the evaluation of suitable existing multi-armed bandit models on our defined scenario. We also present the results obtained for the proposed stylized model developed as part of this work, and provide an objective comparison of the performance of the various bandit models explored to our defined scenario.

In Chapter 5, we present a summary of the main concepts and the key findings obtained as covered in the preceding chapters. Suggestions pertaining to future development of this topic are also provided.

CHAPTER 2:
Literature Review

In this chapter, we provide the reader with a brief introduction to the concept of multi-armed bandits and some of their applications. The other advanced bandit algorithms explored as part of this work are also described in some detail too.

2.1 Classical Multi-armed Bandits Problem

Classical multi-armed bandit problems have been studied by researchers for many decades till date, and have been interwoven together with various practical applications in industry in more recent times. The multi-armed bandit problem relates to the potential challenges involved in applications requiring the use of sequential decision making. The definition of this class of problems was first provided by Thompson [3] way back in the twentieth century. Robbins [4] offered a simple and robust model as a potential solution to the problem in the later years. Operations researchers and mathematicians have not only been fascinated by the original version of the multi-armed bandit problem, but also with the other numerous complex variations arising from the original problem. Considerable effort have thus been spent by numerous researchers on the study of these problems and their related applications over the years till date.

The typical multi-armed bandit problem can be considered to be a classic reinforcement learning problem. The authors in [5] mentioned that these models have in recent times been deployed in a multitude of applications like online web user experience personalization, news recommendations and even in medical clinical trials. These multi-armed bandit models also have a role to play in various machine learning and even artificial intelligence algorithms and applications too.

2.1.1 K-armed bandits problem

The K-armed (multi-armed) bandit problem was first presented by Thompson [3], and a simple overview of the problem definition is presented in this section.

In this problem, a player is given a set of K possible potential actions (or arms which represent the levers on slot machines, or 'bandits'). Let there be a sequence of $T \in \mathbb{N}$ rounds, with each round having time steps indexed by $t = 1, 2, \ldots, T$. In each time step, the player has to select an action from the defined set of K possible actions to execute.

Let $A_t \in \{1, 2, \ldots, K\} \equiv [K]$ represent the player's choice of possible actions. The corresponding selected action $A_t = k$ in each round denoted by $t \in T$ will ultimately enable the player to earn a stochastic reward $X_{k,t} \in \mathbb{R}$. The rewards are drawn from the Bernoulli distribution with unknown means.

Therefore, the ultimate goal of the player would be to maximize the cumulative reward

$$R(T) = \sum_{t=1}^{T} X_{k,t}$$

earned over T rounds, through the optimal choice of action selected in each round.

The optimal selection of action by the player in each round can be considered to be a task that is shrouded in uncertainty and complicated by the fact that the player does not have the advantage of having full information knowledge pertaining to the problem. Each possible action $k \in K$ follows a respective distribution v_k with associated mean p_k. For every single unique action $X_{k,t}$, the associated reward is drawn as an independent, identically distributed sample from the respective Bernoulli distribution v_k. As in all bandit problems and also in real-world practical applications, the player does not possess any knowledge pertaining to both the arms' reward distributions $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ and the true expected reward means $\mathbf{p} = (p_1, p_2, \ldots, p_k)$.

The player desires to achieve maximal expected cumulative reward over time. This is similar to achieving minimum expected cumulative regret over the finite time horizon. The expected regret is a performance measure that quantifies the difference of the expected reward from the best possible arm and the corresponding reward for the selected arm in each round. The expression for the expected regret is given by

$$Reg(T) = Tp^* - \sum_{t=1}^{T} \mathbb{E}(X_{A_t,t}) \quad ,$$

where p^* is the maximal expectation amongst all the various arms and is given by $p^* = \max_{k \in [K]} p_k$. The expected regret serves as a typical performance measure for the various bandit-related algorithms and it is more feasible from an analysis point of view. Thus, maximizing expected cumulative reward in bandit literature corresponds to minimizing expected cumulative regret and both are equivalent performance measures for bandit-related algorithms.

In [5], the authors conclude that it is imperative for the player to select the actions to be executed in a strategic manner. This helps to achieve a long term optimal balance of having actions that contribute to the gathering of knowledge about the unknown distributions and actions that contribute to the earning of best available rewards. These type of actions are broadly classified into exploratory and exploitative actions in literature pertaining to bandit problems. The K-armed bandit problem is an example of a typical problem in which the player has to decide between exploration and exploitation action in each round [5].

2.1.2 Exploration-exploitation dilemma

In numerous sequential decision-making problems like the bandit problem mentioned in Section 2.1.1, there exist an inherent trade-off between deciding on performing exploration or exploitation action, in order to attain the outcome that maximizes the total expected reward over many trials in the long run. The authors in [6] indicated that playing all the respective arms at least once before selecting the arm that pays the best reward amongst all other arms subsequently will tend to result in highly sub-optimal total rewards obtained given the stochastic reward distribution of the individual arms.

The exploration phase allows the player to gain knowledge of the network by learning about the possible sub-optimal actions. This improves the chances of the player selecting the most optimal action to play during the subsequent exploitation phase, but may reduce the total amount of reward obtained. However, if the majority of available time was allocated for exploitation over exploration, there exist the likely possibility of selecting sub-optimal actions to play given the lack of information associated with the unknown reward distributions.

The utilization of optimistic and randomized algorithms allow an appropriate balance between exploration and exploitation phases to be achieved. Optimistic approaches within an algorithm induces exploration actions to be played, which in turn generates optimistic estimates of the possible actions. This follows with exploitation on the most optimal action, by following a greedy policy. Randomization approaches in an algorithm enables the algorithm to learn about different actions much faster than optimistic approaches alone, but trades off on accuracy of the estimates. Thus, the integration of both optimistic and random approaches in algorithms allow a good mix of exploration and exploitation actions to be achieved as part of the attack strategy.

2.2 Review on Multi-armed Bandit Models

This section provides an overview on the models and algorithms that we explored and utilized in this work.

2.2.1 Global confidence bound algorithm

The classical framework presented by the concept of multi-armed bandits can be further extended to other models to address other types of problems. The global confidence bound algorithm presented in [7] was developed based on "combinatorial multi-armed bandits and finite partial monitoring games."

The authors in [7] mentioned that for certain problems involving the use of combinatorial structures, the possible number of arms that can be played "may be exponential to the size of the problem instance." This makes multi-armed bandits less ideal since the running time and regret is exponential to the size of the problem. In addition, multi-armed bandit problems will require feedback on the chosen arm. This feedback may not be available at times. Hence, the global confidence bound algorithm was developed as a unified model to address the issues relating to exponential action space and limited feedback. The performance of the global confidence bound algorithm was evaluated using a "combinatorial stochastic partial monitoring game with linear feedback setting" [7].

In [7], the authors defined a combinatorial stochastic partial monitoring game as a "repeated game between the environment and a player." A summarized version of such a game is

provided here. Let T represent the total time steps played for the game over a finite time horizon. In each discrete time step where $t = 1, 2, \ldots, T$:

- The environment obtains a corresponding outcome vector $\mathbf{v}(t)$ via independent sampling from its predetermined distribution p on $[0, 1]^n$. The environment's outcome vector $\mathbf{v}(t)$ is not known to the player at all times.

- The player selects an action $x(t)$ from a finite action space defined by $\mathcal{X}$ to play. This can be dependent on the feedback received by the player in previous time steps.
- The player then obtains a reward $r\big(x(t), \mathbf{v}(t)\big)$, when action $x(t)$ is applied to the environment outcome vector $\mathbf{v}(t)$. Correspondingly, the player receives feedback $\mathbf{y}(t) = M_{x(t)} \cdot \mathbf{v}(t)$, where M_x is the corresponding transformation matrix given by $M_x \in \mathbb{R}^{m_x \times n}$ whose row dimension m_x is dependent on x. The feedback is so-called linear because $y(t)$ is a linear transformation (given by M_x) of the outcome vector $\mathbf{v}$.

In relation to this thesis, the player represents the attacker and the environment represents the computer network that is being compromised. Once within the network, the attacker (as the player) would first attempt to determine the expected reward associated with the respective nodes by exploring the network (environment) and making intelligent deductions about the order in which the nodes within the network should be compromised, depending on the reward value associated with each network node. This can be achieved by observing the status feedback of the network activity / messages exchanged between the nodes within a network. At every iteration / timestep, the attacker has to decide between exploration of the network, or to attack the best possible identified network nodes. Thus, a regret value (loss value attributed to the fact that the globally optimal set of policy actions was not followed all the time) will be cumulatively formed for every timestep iteration t. It is thus desirable that the model used will enable the attacker to achieve the desired objective of compromising the network with the minimum possible regret value associated with the actions forming the attack strategy at every timestep.

In [8], it is mentioned that multi-armed bandit algorithms in general achieve $O\big(\log T\big)$ distribution-dependent regret and $O\left(\sqrt{T}\right)$ distribution independent regret over T time steps. However, for problems involving the use of combinatorial structures, the available number

of arms which can be played is "exponential to the size of the problem instance" as described in [7]. This makes multi-armed bandit models impractical, and thus requiring combinatorial multi-armed bandit models to be used. In combinatorial multi-armed bandit models, each action played corresponds to a super-arm (which corresponds to a set of base arms). When the super-arm is played, the outcome of all the base arms will be known to the player as feedback [7]. It is also noted by the authors in [7], [9] that combinatorial multi-armed bandit models can achieve the same level of regret as multi-armed bandit models with dependence on the time horizon T.

The proposed global confidence bound algorithm by the researchers in [7] integrates the following concepts from combinatorial multi-armed bandit models and partial monitoring. The researchers in [7] also obtained the corresponding rewards for the set of all possible actions by utilizing a few base arms and respective outcome estimates. This in turn allows the exponentially large action space to be managed. In addition, a global observer set is also used to estimate the rewards of other actions in order to handle the issue of limited feedback [7].

The proposed global confidence bound algorithm by Lin et al. in [7] for "combinatorial partial monitoring games with linear feedback," is able to achieve $O(T^{2/3} \log T)$ distribution-independent regret and also $O(\log T)$ distribution-dependent regret (under the assumption of unique optimal action). The distribution-independent regret and distribution-dependent regret bounds for the global confidence bound algorithm differs from the corresponding regret bounds for the multi-armed bandit problem presented earlier. The global confidence bound algorithm accounts for limited bandit feedback, while the multi-armed bandit prob-lem assumes that full information feedback is available, and the global confidence bound algorithm regret bounds indicate that learning with limited feedback is significantly harder than in the multi-armed bandit problem instance.

In addition, the authors in [7] highlighted that the regret bounds for the global confidence bound algorithm have a linear dependency on $\log |\mathcal{X}|$, where $\mathcal{X}$ is the size of the action space available to the player. This in turn suggests that the player can possibly avoid explicit enumeration of all available actions in order to learn about the rewards that the environment might potentially offer for each action that the player might take [7].

2.2.2 Stochastic online learning algorithm

Li et al. in [10] mentioned that stochastic online learning can be classified as part of the general framework for a sequential decision problem. In each timestep, the player selects an action to play from a given finite set of possible actions and thus receives some random corresponding reward with some random feedback observations obtained. The feedback model could be a full information feedback (which is unrealistic) model in which the random rewards of all actions can be observed regardless of the action played, or it could possibly be similar to the bandit feedback model in which the player is able to observe the random reward associated with the selected action played [10].

The authors in [10] considered the "problem of stochastic online learning with general probabilistic graph feedback", in which the respective directed edges in the feedback graph have an associated probability P_{ij}. Two main problem types were studied by the authors in this work, namely the one-step case and the cascade case. In the one-step case, the player would be able to observe a sample reward feedback of an arm j with independent probability P_{ij} after selecting an arm i to play. In the cascade case, after the player selects arm i to play, he would be able to observe the feedback of all available arms j as part of a probabilistic cascade starting from i for each edge (i, j) with respective probabilities P_{ij}. Consequently, if an arm i is selected to be played or observed then a corresponding reward sample of arm j would then be observed with an independent probability P_{ij}. The authors then further analyzed the asymptotic lower bounds for the regret values in both of these cases and designed algorithms which had regret upper bounds that corresponded to the asymptotic lower bounds with high probability in this case [10].

In general, this algorithm is able to achieve $O(\log T)$ regret bound for the different cases investigated by the researchers in [10], which is rather similar that of the multi-armed bandit problem presented earlier. However, the regret bound expressions derived and presented in their work are gap-dependent bounds [10].

Interestingly, the model presented in this algorithm is largely similar in some aspects and might be considered as a subset of the model proposed in the global confidence bound algorithm. Both models incorporate the general idea of attempting to obtain as accurate estimates of the unknown reward means for the arms in the exploration rounds. In addition, both models rely on the notion of evaluating the reward gap between the best two arms

identified so far as a condition for performing exploitation on the best arm. Exploitation on the best arm identified is only executed when the reward gap is significantly large, to minimize possible regret by playing the various sub-optimal arms. However, the model in the global confidence bound algorithm is also more conservative as it incorporates an additional condition of transiting into exploitation when sufficiently large number of exploration rounds have been played. This helps the model in the global confidence bound algorithm from performing endless repetitions on exploration if there are multiple optimal actions present, as compared to the model presented in this particular algorithm.

2.3 Review on Multi-armed Bandits in Cyber Models

This section provides an overview of how some of the various multi-armed bandit models have been deployed for use in the cybersecurity domain, and is by no means extensive given the ever-dynamically changing nature of the issues to address within this domain.

Firstly, Elderman et al. in [11] considered the use of a Markov game with incomplete information and stochastic elements to model cyber-security simulation attacks on a network. Such games are typically used in the study of multi-agent reinforcement learning problems, where agents are unable to observe the reward functions of other agents but instead learn via interactions between different agents. The researchers applied this model as an attacker-defender stochastic zero-sum game, where the stochasticity emanates from a detection parameter representing the probability of detection in the event that an attack on the network fails. This simulates real scenarios in which attacks on networks may go unnoticed if detection occurrences are infrequent. The objective of the attacker in this case would be to determine a path through the network in order to penetrate and reach the high-valued nodes. The attacker is also able to use reinforcement learning techniques to dynamically adapt to the defense strategy adopted by the defender, while the defender would also be able to dynamically allocate defense resources along the intermediate nodes along the attacker's intended path. The authors examined and proposed various attack and defense strategies that is based on approximate dynamic programming, which would be very much suitable when the game is repeatedly played.

Secondly, Qian et al. in [12] examined a pertinent problem often encountered by the defenders, which is the number of assets and nodes within a network might be too vast to

simultaneously protect in the event of a massive attack. In the model proposed by the authors, the intensity of attacks for the unprotected assets vary according to a Markovian process with an unknown transition matrix, while the intensity of attacks for the protected assets also vary in accordance with another Markovian process following a different transition matrix. As such, the authors proposed the use of restless multi-armed bandit models as a suitable basis for the defender to achieve an optimal allocation of finite defense resources that are available for deployment.

Thirdly, Liu and Zhao in [13] studied an example of a large-scale cyber network comprising of various network components that are either in a healthy state or an abnormal state. The states of the various components could transit between these states over time in accordance with an arbitrary stochastic process. The authors considered this to be related to a practical application of a typical Intrusion Detection System (IDS) of a network, where the objective is to locate components in an abnormal state in the shortest possible amount of time. As such, the authors devised a dynamic component probing strategy using a restless multi-armed bandit model that minimizes the long-term cost incurred by components being in the abnormal state. This is done by the model learning from past decisions and observations from the environment, and then dynamically changing the probing actions taken.

Jung et al. in [14] studied and proposed that contextual multi-armed bandit models could potentially be utilized for the design of self-learning security modules for computer networks and other related tasks. In this work, the authors studied how the contextual multi-armed bandit model can be formulated as a one-shot sequential learning problem to prevent HTTP-based attacks on web servers as a potential practical application. Next, the authors also designed an algorithm that was computationally cheap for the purpose of general contextual multi-armed bandit learning that would be able to target specific domains using finite countable actions. The designed algorithm can be designed as a complete self-learning meta filter for web servers that is not dependent on feedback from end-users. Hence, such meta-filters would be capable of integrating several existing security measures used in the cyber defense of web servers, and does so using only stochastic and sparse feedback data.

Gutierrez et al. in [15] considered and studied the problem of how humans learn in an interactive environment in the context of a multi-armed bandit setting. The authors considered an adversarial, contextual multi-armed bandit scenario and studied a potential attacker-defender

model which was motivated by the area of deception that is used in the cyber-security domain. In this typical scenario, the network defenders would deploy deceptive decoys as part of the network's defense and the attackers had to learn how to avoid them to exploit the actual high-valued nodes in the network. The authors determined that an adaptive defensive strategy had the best performance against the human cyber attackers, and presented the most challenge for the human attackers to learn how to circumvent the defensive strategy deployed by the defenders. Thus, the use of the multi-armed bandit models as part of the attacker-defender model would enable more robust network defenses to be developed and deployed in order to actively mitigate the effectiveness of potential attackers learning and devising ways to penetrate through the network defenses.

Our work in this thesis aims to contribute to this area of research by proposing suggested approaches for formulating optimal network attack strategies given uncertainties relating to network topologies as well as the associated rewards for the various network nodes. This allows effective and optimized network attack strategies to be devised, which help improve the robustness of present and future network defense systems.

CHAPTER 3:
Problem Formulation and Methodology

For our problem, we consider the application of a typical Local Area Network (LAN) setup. Such a setup is deployed in practically almost all computer network topologies of various sizes, with various computing entities represented as network nodes being interconnected via a central LAN node such as a network switch. Besides enabling multiple hosts devices (nodes) to be interconnected in a LAN, a network switch also performs key functions like forwarding of messages to specified network hosts and performing data traffic and bandwidth management within the LAN. An illustration of a simple 'star' LAN network topology with a single switch and K number of nodes is as shown in Figure 3.1.

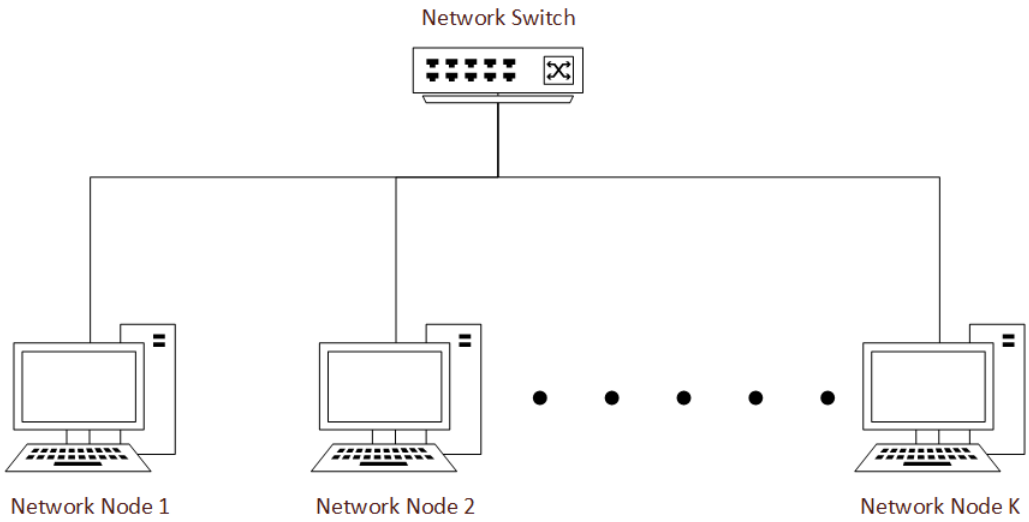

Figure 3.1. Typical Local Area Network Topology, with $\mathcal{K}$ Nodes.

An attacker who has managed to surreptitiously gain unauthorized access into the network, wishes to exploit the most valued/important node to disrupt network operations. He is only aware about the existence of the network switch and some connected $\mathcal{K}$ network nodes, and no other network topology related information. However, he has no prior knowledge about the "value" of the individually connected nodes (arms). Thus, the attacker would need to devise an optimal plan of action of either "exploration" (observing the data exchange

through the switch) or "exploitation" (attacking the individual nodes). This is achieved by the attacker performing sequential interactions with the switch and the other nodes within the network in optimal fashion over the finite time horizon.

If the attacker decides to exploit an individual node, he gains a new observation on that single node as corresponding feedback. These observations obtained over time enable the attacker to estimate the importance value associated with respective nodes, and focus on disrupting the operation of the most important node by attacking it.

If the attacker decides to explore the network switch instead, he obtains as intelligence feedback the corresponding observations on a random multiple number of connected network nodes each time. These observations obtained enables the attacker to identify the promising nodes to exploit over time, as part of the attack strategy.

3.1 Background

The multi-armed bandit framework reviewed in Chapter 2 provides a possible optimal strategy for the attacker to sequentially interact with the various network nodes over time. Through this sequential interaction process, he obtains possible intelligence feedback to help identify the more promising nodes to attack within the finite time horizon.

Let $[\mathcal{K}] = \{1, \ldots, \mathcal{K}\}$ represent the set of arms that correspond to each of the network nodes. The index $\mathcal{K} + 1$ represents the super-arm, which corresponds to the switch within the network. Sampling an arm corresponds to attacking an individual node in our problem.

When sampling an individual arm $k \in [\mathcal{K}]$, the reward obtained would be the associated node importance value given by the respective Bernoulli distribution parameter p_k. The distribution and value of parameter p_k is assumed to be unknown; otherwise the attacker would just focus on the arm with the largest p_k. Instead, the attacker obtains as intelligence feedback an observation on the sampled arm which is manifested as an independent Bernoulli random variable with parameter p_k. In case the arm selected by the attacker in a particular iteration is not the optimal arm, the difference in expected reward values between the sampled arm and the optimal arm will contribute to the cumulative expected regret obtained over time.

If the attacker decides to sample the super-arm, the resultant reward is zero because the switch does not manifest intelligence about itself in our defined scenario. Hence, the attacker incurs a regret equal to the magnitude of the reward value associated with the optimal arm. However, the attacker will be able to observe a "**0**" (no intelligence value), a "**1**" (positive intelligence value), or a "**-1**" (no information can be inferred due to noise) for a random subset of the k individual arms, independently of each other.

The parameter η gives the probability with which each arm yields an independent observation when the super-arm is pulled by the attacker, and models the effect of observing the network switch in practice. The network switch facilitates the flow of data exchange amongst the network nodes, and the high volume of network data flow in every time step only allows the attacker to obtain observations for a subset of arms realistically in a single time interval.

For simplification of the model, we assume that for each arm $k \in [\mathcal{K}]$:

$$\begin{aligned} \text{P(intelligence value = -1)} &= 1 - \eta \\ \text{P(intelligence value = 0)} &= \eta(1 - p_k) \\ \text{P(intelligence value = 1)} &= \eta(p_k) \end{aligned}$$

In each round, the attacker makes a decision on whether to either sample from any of the k individual arms, or to sample from the super-arm. When the attacker samples an individual arm k, he receives as information the outcome (a "**0**" or "**1**", which serves as useful information to determine the good arms), and the corresponding reward (of "**1**" with probability p_k, and "**0**" with probability $1 - p_k$).

If the attacker decides to sample the super-arm (denoted by index $\mathcal{K} + 1$), no reward is collected. However, he potentially receives information from several arms, since each arm has a chance η of returning a "**0**" or "**1**" value. This information can then be utilized by the attacker to identify the potentially promising arms from the set of $\mathcal{K}$ arms in the network that should be sampled to obtain maximum possible reward over time.

As earlier mentioned, if the attacker samples an arm k, he earns a reward given by the associated node importance value which is given by Bernoulli distribution parameter p_k.

He obtains as intelligence feedback, an observation on the arm manifested as a Bernoulli random variable with parameter p_k. If the super-arm is sampled, the attacker earns no corresponding reward but instead obtain intelligence feedback on a random number of arms.

Clearly, it can be observed that when $\eta \gg 1/\mathcal{K}$, the attacker should initially pull the super-arm since it collects intelligence from many more than one arm (in expectation), albeit without earning any potential rewards. The attacker should then focus on the more promising arms when they have been identified, in order to maximize total potential reward earned, in expectation. When $\eta > 1/\mathcal{K}$, the attacker should divide the initial number of pulls between the super-arm and the other arms since the intelligence feedback on multiple arms from the super-arm is limited. If $\eta \leq 1/\mathcal{K}$, then the attacker should ideally allocate the initial number of pulls on the other arms instead of the super-arm since almost no useful intelligence on multiple arms can be collected under this setting when the super-arm is pulled.

The main research question then, is: What should be the optimal plan of action for exploration and exploitation of the network that should potentially be adopted by the attacker? From an operational perspective, this would be beneficial in identifying potential gaps in system defenses and thus translates into improving the robustness of these defense mechanisms. We consider three different algorithms in turn as part of this Chapter, and compared their individual cumulative regret performance for our defined scenario presented earlier.

3.2 The Global Confidence Bound Learning Algorithm

The global confidence bound learning algorithm presented in [7] for a "combinatorial partial monitoring game with linear feedback setting" is applicable to our research problem.

3.2.1 Definition of parameters

The parameters used in the global confidence bound algorithm is defined as follow.

- ν: the vector of true means of the unknown reward distribution for the set of arms.
- $\mathcal{X}$: the set of K possible actions/arms, including the super-arm.

- $\hat{x}$: an individual arm that provides the best expected reward on $\hat{\nu}$.
- $\hat{x}^{-}$: an individual arm that provides the second best expected reward on $\hat{\nu}$.
- σ: the observer set of actions, where the attacker plays a certain action in each time step. $\sigma = \{x_1, x_2, \ldots, x_{K+1}\}$, comprising of the K arms and the super-arm.
- α: a parameter used in the global confidence bound. $\alpha = 24 \times K$ in our analysis.
- $\tilde{\nu}(t)$: vector of estimated outcomes at time t. Note that $\mathbb{E}[\tilde{\nu}(t)] = \nu$.
- $\hat{\nu}$: cumulative average of estimated outcomes, $\tilde{\nu} = \frac{1}{n_\sigma} \sum_{j=1}^{n_\sigma} \tilde{\nu}_j$.
- $f_{\mathcal{X}}(t)$: frequency function used in the global confidence bound. $f_{\mathcal{X}}(t) = \ln t + 2 \ln |\mathcal{X}|$
- n_σ: the number of exploration rounds played so far.
- $\overline{r}(x, \cdot)$: the corresponding reward obtained by playing action x.
- $\mathcal{R}$: the reward matrix consisting of all possible reward vectors
- M_x: linear feedback transformation matrix associated with arm x, where $x \in \mathcal{X}$.

3.2.2 Pseudocode of the global confidence bound algorithm

The pseudocode of the global confidence bound algorithm in [7] is presented next.

Algorithm 1: Global Confidence Bound Algorithm

Require: $\sigma,\ \alpha,\ f_X(t); \forall x \in X,\ \overline{r}(x,\cdot),\ M_x$

Initialize t $\leftarrow 0$, $n_\sigma \leftarrow 0$;
loop
  if $t = 0$ **then**
    state $\leftarrow$ begin_exploration {initialize $\hat{v}$}
  else
    $\hat{x} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, \overline{r}(x, \hat{v})$
    $\hat{x}^- = \underset{x \in \mathcal{X} \setminus \{\hat{x}\}}{\operatorname{argmax}}\, \overline{r}(x, \hat{v})$
    if $\left(\overline{r}(\hat{x}, \hat{v}) - \overline{r}(\hat{x}^-, \hat{v}) > \sqrt{\frac{\alpha f_X(t)}{n_\alpha}} \right)$ **or** $\left(n_\sigma > t^{\frac{2}{3}} f_X(t) \right)$ **then**
      state $\leftarrow$ exploitation
    else
      state $\leftarrow$ begin_exploration
    end
  end
  if *state = begin_exploration* **then**
    {exploration phase:}
    for $s \leftarrow 1;\ s \leq |\sigma|;\ s \leftarrow s + 1$ **do**
      play x_s in observer set σ, and observe y_s
      if $s = |\sigma|$ **then**
        $n_\sigma \leftarrow n_\sigma + 1$
        $\vec{y}_{n_\sigma} = \left(y_1; y_2; \cdots; y_{|\sigma|}\right)$
        $\tilde{v}_{n_\sigma} = \mathbf{I}\left(M_\sigma, \vec{y}_{n_\sigma}\right)$ {estimate outcomes}
        $\hat{v} = \frac{1}{n_\sigma} \sum_{n_\sigma}^{j=1} \tilde{v}_j$ {take average}
      end
      t $\leftarrow$ t + 1
      state $\leftarrow$ in_exploration
    end
  else
    {exploitation phase:}
    play action $\hat{x}$
    t $\leftarrow$ t + 1
  end
end loop

3.2.3 The global confidence bound algorithm model

The model presented in [7] can be categorized as a "combinatorial partial monitoring game with linear feedback", that involves a repeated game with the attacker and the environment in discrete time steps. This model is suitable for our application, where in each time step the attacker has to select an action to take that either involves exploration of the network or the exploitation of a particular node in the network. Thus, based on the action selected by the attacker, a corresponding reward value is obtained and the attacker needs to have an optimal action plan that optimizes his total expected reward earned over time. An optimal balance of exploratory actions and exploitation is desired to be achieved in this case, which is the goal of the global confidence bound algorithm detailed in [7].

Prior to the commencement of the game, an n-th dimensional fixed probability distribution in $[0, 1]^n$ is determined by the environment. This forms the reward distribution of the true means associated with each arm (network node), and is not revealed to the attacker throughout. This typically depicts the scenario in real-world applications, where the attackers typically do not have knowledge on which connected target node is the most rewarding to attack.

A fixed time horizon T is defined for algorithm execution, and represents the finite amount of time that the attacker has before being detected by defense mechanisms of the network. In each discrete time step t, the environment performs a sampling of an independent random vector $v(t) \in [0, 1]^n$ from the fixed probability distribution. Both $v(t)$ and the predetermined probability distribution are not known to the attacker throughout. This represents the outcome of the environment in that particular time step, and the outcome changes between time steps.

At each time step, the attacker selects an arm $x(t) \in \mathcal{X}$ to pull. $\mathcal{X}$ consists of the set of K arms and the super-arm. Thus, when $x(t)$ is applied to the corresponding environment outcome vector $v(t)$, the attacker gains an associated reward $r\big(x(t), v(t)\big) \in \mathcal{R}$. The attacker is also able to receive a linear feedback $y(t)$, where $y(t) = M_{x(t)} \cdot v(t)$ is the inner product of the $x(t)$ row of the matrix $\mathbf{M}$ and the outcome vector $\mathbf{v}(t)$.

Similar in concept to the majority of bandit algorithms, the proposed global confidence bound algorithm in [7] alternates between exploration and exploitation phases as time progresses. The empirical mean of the outcome vector $\tilde{v}$ is initialized by conducting one round of exploration when the algorithm first executes, as given by line 4 in algorithm 1. After which, the global confidence bound algorithm computes the global actions $\hat{x}$ and $\hat{x}^{-}$ that produces the best expected reward and next best expected reward respectively in lines 6 and 7 of algorithm 1. There is no enumeration for the set of all possible actions [7].

The global confidence bound, $\sqrt{\frac{\alpha f_X(t)}{n_\alpha}}$ implemented in line 8 of algorithm 1 is used to determine whether an exploration or exploitation action should be performed in each time step, where $\alpha = 24 \times K$. It is also used to evaluate if there is significant separation (distinction) between the globally optimal and second optimal solutions. If there is one, then it would be optimal to perform exploitation on the current identified optimal action $\tilde{x}$, given by line 9 in algorithm 1. Hence, the global confidence bound algorithm in [7] exchanges regret from additional exploration rounds with reduced error probability during exploitation to avoid potentially incurring a high regret cost that would be linear to the potential space size of all possible actions.

From [7], in the event that the difference of the estimated expected reward of the best and second best actions remains very small (meaning that the reward gap between best two actions is insignificant), the global confidence bound algorithm avoids continuous repetition of the exploration phase and commences exploitative action on the best action identified if adequate exploration rounds have been performed. This implementation can be found in line 8 of algorithm 1. The global confidence bound algorithm refrains from evaluating differences in expected reward between all pairs of possible actions [7].

3.2.4 Application of global confidence bound algorithm to our problem

Utilizing the notation presented in [7], let the support for the outcome vector $\tilde{\nu}(t)$ be a $2K$ dimensional vector of 0's or 1's for the first K entries and 0's, 1's and -1's for the last K entries. K denotes the total number of arms(network nodes).

$$(\overbrace{\underbrace{\{0,1\}}_{\text{arm 1}}}^{v_1};\overbrace{\underbrace{\{0,1\}}_{\text{arm 2}}}^{v_2};\ldots;\overbrace{\underbrace{\{0,1\}}_{\text{arm K}}}^{v_K};\overbrace{\underbrace{\{0,1,-1\}}_{\text{arm 1}};\ldots;\underbrace{\{0,1,-1\}}_{\text{arm K}}}^{v_{K+1}\quad\quad v_{2K}}}^{\text{super arm}})\quad .$$

Thus, the environment chooses a value of either 0 or 1 for the individual arms 1 through K, and either a value of 0,1, or -1 for each of the corresponding individual arms when the super-arm is sampled. Hence, the distribution of $\tilde{\nu}(t)$ is the product of $2K$ distributions. The first K elements are Bernoulli random variables $\big(Ber(p_1),\ldots,Ber(p_k)\big)$ and the last K elements have the following distribution for $k \in (1,\ldots,K)$:

$$\begin{aligned}
\mathrm{P}\big(\nu_{K+k} = \text{-}1\big) &= 1-\eta && \Longrightarrow \text{ intelligence value = -1} \quad ,\\
\mathrm{P}\big(\nu_{K+k} = 0\big) &= (1-p_k)\eta && \Longrightarrow \text{ intelligence value = 0} \quad ,\\
\mathrm{P}\big(\nu_{K+k} = 1\big) &= (p_k)\eta && \Longrightarrow \text{ intelligence value = 1} \quad .
\end{aligned}$$

When sampling an individual arm $k \in [K]$, the transformation matrix is a $2K$-diemnsional vector $M_k = e_k$ with a 1 in the k'th position and zeros elsewhere. Thus, $y(t) = M_k \cdot \tilde{\nu}(t) \in \{0,1\}$ when $\hat{x}(t) = \mathrm{k}$, for $k \in [K]$. Hence, $y(t)$ returns the output value of arm k.

The resultant transformation matrix when the super-arm is sampled is given by

$$M_{K+1} = (\mathbf{0} \quad \mathbf{I}) \quad ,$$

where $\mathbf{0}$ is the $K \times K$ zero matrix, and $\mathbf{I}$ is the $K \times K$ identity matrix.

The feedback $y(t) = M_{x(t)} \cdot \hat{v}(t)$ is the vector as shown below when $x(t) = K + 1$

$$(\underbrace{\{0, 1, -1\}}_{\text{outcome of arm 1}} ; \ldots ; \underbrace{\{0, 1, -1\}}_{\text{outcome of arm K}}) .$$

In addition, $\sigma = \{1, \ldots, K, K + 1\}$ and $|\sigma| = K + 1$. The matrix M_σ is the $2K \times 2K$ identity matrix

$$M_\sigma = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} ,$$

where rank $M_\sigma = 2K = n$.

The expected reward, as a result from sampling of either of the individual k-th arms or the super-arm, can be expressed as $r(x(t), \tilde{v}(t))$.

For $x(t) = k \in [K]$,

$$r(k, \tilde{v}(t)) = y(t) = M_k \cdot \tilde{v}(t) = \text{outcome of arm k} . \tag{3.1}$$

For $x(t) = K + 1$,

$$r(K + 1, \tilde{v}(t)) = 0 . \tag{3.2}$$

Thus, when the super-arm is sampled, the corresponding expected reward gained by the attacker is zero. Hence, it can be observed that the reward function, r, is a linear function of $\tilde{v}$.

As with all typical bandit models, the fundamental performance metric would be to define and determine the expected cumulative regret over the specified fixed time horizon T. It is in the interest of the attacker to achieve minimum expected regret (i.e. maximize expected

reward) at the end of the game at time horizon T. Thus, the performance of various bandit models can be evaluated using the plot of the cumulative regret bounds and comparing the regret curves for the over time.

3.2.5 Cumulative regret bound of the global confidence bound algorithm

As in [7], the expected regret over the finite time horizon T for the global confidence bound algorithm can be defined as follow.

$$R(T) = T \max_{x \in \mathcal{X}} \mathbb{E}\Big[r(x, v)\Big] - \sum_{t=1}^{T} \mathbb{E}\Big[r\big(x(t), v(t)\big)\Big] \tag{3.3}$$

In equation (3.3), the first term represents the corresponding reward that is obtained by the attacker if he pulls the arm that yields the maximum expected reward for each time step. The associated reward is taken from the predetermined probability distribution at the start of the game, and is unknown to the attacker throughout. In our application, if the attacker decides to pull the super-arm, then no positive reward is obtained. Instead, the attacker obtains a regret corresponding to the reward associated with the optimal arm.

Define $\nu = \mathbb{E}[v]$ as the vector of expected values of v and let $\bar{r}(x, \nu) = \mathbb{E}[r(x, v)]$. It is further assumed that the expected reward term is a function of x and the expected outcome vector ν of v. It is mentioned in [7] that this assumption is satisfied when the corresponding reward function is a linear function of v, or if the distribution of v can be determined by the mean vector of the distribution ν. In our analysis, the draw the unknown reward distribution of the means as independent Bernoulli random variables and hence this assumption is satisfied.

We let x^* be the expected reward associated with the best optimal arm under the expected outcome vector ν. Thus, $x^* = \max_{x \in \mathcal{X}} \bar{r}(x, \nu)$, and the expected regret function can be written as follows,

$$R(T) = T \cdot \bar{r}(x^*, \nu) - \sum_{t=1}^{T} \mathbb{E}\Big[\bar{r}\big(x(t), \nu\big)\Big] \quad . \tag{3.4}$$

In addition, the regret bounds on v for the global confidence bound algorithm can be formulated into distribution-independent or distribution-dependent bounds.

The distribution-independent bound for the global confidence bound algorithm in [7] is

$$R(T) \leq R_{max}|\sigma| \cdot T^{\frac{2}{3}}\left(\ln T + 2\ln|\mathcal{X}|\right) + \frac{8}{3}L\beta_\sigma T^{\frac{2}{3}} + R_{max}\left(|\sigma| + \frac{4e^2}{|\mathcal{X}|^4}\right) \quad , \tag{3.5}$$

where L is a Lipschitz constant and β_σ is a constant of the instance of the game that is independent of the outcome distribution p.

In addition, the distribution-dependent bound for the global confidence bound algorithm for an outcome distribution of p and with mean outcome vector ν is

$$R(T) \leq \sum_{x\in\sigma} \Delta_x \cdot \left[\frac{96L^2\beta_\sigma^2}{\Delta_{min}^2}\left(\ln T + 2\ln|\mathcal{X}|\right) + \frac{4e^2}{|\mathcal{X}|^4}\ln T + 1\right] + \Delta_{max}\cdot\left(\frac{3e^2}{|\mathcal{X}|^4} + \frac{941L^3\beta_\sigma^3}{\Delta_{min}^3}\right) \quad , \tag{3.6}$$

where

$$\begin{aligned}\Delta_x &= \bar{r}(x^*, v) - \bar{r}(x, v)\\ \Delta_{max} &= \max\{\Delta_x : x \in \mathcal{X}\}\\ \Delta_{min} &= \min\{\Delta_x : x \in \mathcal{X}, \Delta_x > 0\} \quad .\end{aligned}$$

From equation (3.5), the global confidence bound algorithm is able to achieve an $O\left(T^{\frac{2}{3}}\ln T\right)$ distribution-independent regret that is rather close to the $O\left(T^{\frac{2}{3}}\right)$ theoretical regret bound for partial monitoring games [7].

As mentioned in [7], ff a particular instance of the game has a unique optimal arm under the outcome distribution p and mean outcome vector ν, then the global confidence bound algorithm is able to achieve $O\left(\log T\right)$ distribution-dependent regret which in turn corresponds to the theoretical lower bound for classical multi-armed bandit problems .

3.3 LiEtAl Algorithm for Stochastic Online Learning

While the global confidence bound algorithm presented in [7] trades off the initial regret from exploration for accuracy in exploiting the best arm in the later time steps, the LiEtAl algorithm in [10] differs slightly in its approach. From [10], the LiEtAl algorithm constantly evaluates both the accuracy of the estimates of the arms' reward mean estimates and the reward gaps between arms to determine the corresponding action to take in each time step.

The LiEtAl algorithm presented in [10] was developed in consideration of the problem of stochastic online learning, where it is utilized as a generic framework for addressing sequential decision problems. These type of problems are very similar to the typical setup of problems applicable to bandit models as described in earlier sections. In each time step within a defined time horizon, the attacker plays an action from the available set of all possible actions. He in turn obtains a random associated reward and also obtains some random feedback for the chosen action [10].

From [10], there are various types of feedback models possible. The full feedback model enables the attacker to learn over time the random rewards associated with all possible actions regardless of which action was chosen. The typical bandit model feedback on the other hand, only enables the attacker to learn only about the associated reward for a chosen particular action [10]. These feedback models can be generalized and characterized using the concept of directed graphs [16]. In such graphs, each edge (i, j) corresponds to an observation for the random reward associated with action j to the attacker, if action i is played.

3.3.1 Model setting

Specifically, the authors in [10] considered the stochastic learning problem with general probabilistic graph feedback, and the notation used here differs slightly from that of the global confidence bound algorithm in [7] as presented in the earlier section.

In the feedback graph, the directed edges in the graph each has an associated probability p_{ij}. From [10], the problem considered can be characterized by a quadruple (V, E, p, μ). $V = [K]$ represents the finite set of all possible actions. $E \subseteq V \times V$ denotes the set of directed edges between different actions. $p : E \rightarrow (0, 1]$ provides the mapping of the directed edges to the associated triggering probabilities. $\mu = \{\mu_i\}_{i \in V}$ contains the reward

distributions of all possible actions. Without any loss of generality, it is assumed that each reward distribution candidate is also 1-sub Gaussian. This is trivially satisfied by the Bernoulli reward distributions which we later consider in our analysis.

As presented in [10], the directed probabilistic feedback graph can be denoted by $G = (V, E, p)$. It is assumed that the attacker has knowledge about G and that the μ_i's have 1-sub Gaussian tail. However, he does not have knowledge on the underlying reward means θ_i's. In each time step, a corresponding reward vector $r_t = \big(r_t(i) : i \in V\big)$ is drawn by the environment by performing independent sampling of $r_t(i) \sim \mu_i$. A random graph $G_t = \big(V, E_t\big)$ based on G is also generated in each time step. Also, $E_t = \{(i, j) \in E : o_{tij} = 1\} \subseteq E$, and o_{tij} is an independent Bernoulli random variable with associated mean p_{ij}.

Correspondingly, the attacker selects an action $i_t \in V$ in each time step, using all prior observations collected about the various arms so far. This is done without observation of either G_t or r_t in every time step. Thus, an instantaneous reward $r_t(i_t)$ is received by the attacker in each time step.

3.3.2 Key model assumptions

There are several assumptions made with regards to the algorithms presented in [10], and these are listed as follow. These assumptions are made by the authors in [10] so that the corresponding reward associated with the selected action is observable.

1. The feedback graph G is observable, where each action has an opportunity to be observed by pulling the corresponding arm. That is, for each possible action (arm) j, there exist an edge $(i, j) \in E$ for some i.

2. Each feasible distribution vector is composed of corresponding reward distributions of the "same type".
   Denote C as the set of all possible reward distributions. Denote $\mathcal{S}$ as the set containing all feasible vectors of reward distributions. Denote KL as the Kullback–Leibler (KL) divergence. Thus, for every $\mu \in \mathcal{S}$, $\mathrm{KL}(\mu_i, \mu_j)$ is well-defined for any $i, j \in V$. If $\mathrm{KL}(\mu_i, \mu_j)$ is well-defined for any $i, j \in V$ for every $\mu \in C^V$, then $\mu \in \mathcal{S}$.

3. The KL divergence of the underlying arms' reward distributions is continuous with respect to their corresponding mean reward values. Let there exist some universal constant $B > 0$ such that for each $\mu_i, \mu_j \in C$ and any $\epsilon \in (0, 1)$, there exists $\mu_i^{'} \in C$ which satisfies $\mathrm{KL}(\mu_j, \mu_i^{'})$ is well-defined, $\theta(\mu_i^{'}) + \epsilon \leq \theta(\mu_i) < \theta(\mu_i) + 2\epsilon$ and $|\mathrm{KL}(\mu_j, \mu_i^{'}) - \mathrm{KL}(\mu_j, \mu_i)| \leq B\epsilon$.

3.3.3 Definition of algorithm parameters

The algorithm presented in [10] required the use of the following parameters, and the notation used differs across the various algorithms explored.

- η, the probability with which each arm yields an observation from the super-arm.
- β, an auxiliary function that is any non-decreasing function satisfying $0 \leq \beta(n) \leq n/2$ and the subadditivity $\beta(m + n) \leq \beta(m) + \beta(n)$.
- p_k, the true expected mean value of the k^{th} arm.
- $X_{k,t}$, the estimated sample mean of the reward for the individual $k - th$ arm at time t. Also denoted as $\hat{\mu}_{t,k}$.
- $\hat{\mu^*}_t$, the arm with the largest mean estimate amongst all the arms in round t.
- $A_s \in \{1, 2, \ldots, K + 1\}$, where A_s is a random variable that indicates which arm was chosen by the algorithm in round $s = 1, 2, \ldots, T$. If $A_s = K + 1$, then it indicates that the super-arm was chosen in that round.
- $N_k(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = k\}, \forall k \in \{1, \ldots, K + 1\}$, where $N_k(t)$ represents the total number of times an individual arm was pulled.
- $n_k(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = K + 1, X_{k,t} \text{ observed}\}, \forall k \in \{1, \ldots, K\}$, where $n_k(t)$ represents the number of observations of the $k - th$ arm obtained from pulling the super-arm.
- $m_k(t) = n_k(t) + N_k(t), \forall k \in \{1, \ldots, K\}$, and $m_{K+1}(t) = N_k(t)$.
- $N^e(t)$ represents the number of exploration rounds performed up to time t, $N^e(0) = 0$.

3.3.4 Pseudocode of the LiEtAl algorithm

The LiEtAl algorithm in [10] is more complex in the general setting, as presented in our earlier analysis and discussion. Hence, we adapted the LiEtAl algorithm in [10] to our defined problem and the pseudocode of the adapted algorithm is presented next.

Algorithm 2: LiEtAl Algorithm

1 Check whether there exists an arm $k \in \{1, \ldots, K\}$ such that $m_k(t-1) < 0.5\eta N_{K+1}(t-1)$. If so, denote this arm k' and proceed to step 2. If not, proceed to step 3.

2 With probability 0.5 play arm k' and with probability 0.5 play the super-arm. Record the observations. Set $N^e(t) = N^e(t-1)$ and proceed to the next time point $t+1$ (i.e. skip steps 3 - 7)

3 For each arm $k = \{1, \ldots, K+1\}$ compute $C_k(t) = N_k(t)/16 \log(t-1)$ and for arms $k = \{1, \ldots, K\}$ compute the mean estimate $\hat{\mu}_{t-1,k}$ based on previously observed data. Let $\hat{\mu^*}_{t-1} = \max_{k=1,\ldots,K} \hat{\mu}_{t-1,k}$. For each arm $k \in \{1, \ldots, K\}$ except the one with the largest mean estimate ($\hat{\mu^*}_{t-1}$) check the following condition

$$C_k(t) + C_{K+1}(t)\eta \geq \frac{1}{\hat{\mu}_{t-1,k} \log\left(\frac{\hat{\mu}_{t-1,k}}{\hat{\mu^*}_{t-1}}\right) + (1-\hat{\mu}_{t-1,k}) \log\left(\frac{1-\hat{\mu}_{t-1,k}}{1-\hat{\mu^*}_{t-1}}\right)}$$

Store any arms for which the condition does not hold in set D_t. If D_t is empty, proceed to step 4, if not, proceed to step 5.

4 Play the arm $k \in \{1, \ldots, K\}$ with the largest mean estimate. Record the observation, set $N^e(t) = N^e(t-1)$ and proceed to the next time point $t+1$ (i.e. skip steps 5 - 7).

5 Check whether there exists an arm $k \in \{1, \ldots, K\}$ such that $\eta m_k(t-1) < 2\beta N^e(t-1)/(K+1)$. If so, denote this arm k', and proceed to step 6. If not, proceed to step 7.

6 With probability 0.5 play arm k' and with probability 0.5 play the super-arm. Record the observations, set $N^e(t) = N^e(t-1) + 1$ and proceed to the next time point $t+1$ (i.e. skip step 7).

7 Play an arm sampled uniformly at random from the set D_t. Record the observations, set $N^e(t) = N^e(t-1) + 1$ and proceed to the next time point $t+1$.

3.3.5 Description of the algorithm pseudocode

The algorithm begins execution as follows. It initializes various key quantities, in particular the number of exploration rounds played N^e, and also the estimated mean averages for the various arms. Each exploration round played increments the value of quantity N^e by 1, and presents an opportunity for the attacker to learn more about the unknown reward mean distribution associated with the arms. Hence, at the end of the algorithm run, we would expect that the attacker has a better understanding of the true reward means for the various arms, and thus the estimated mean averages will serve as reasonable estimates of the true mean values.

For every time step t, if for some action j the cumulative number of observations is less than half the total expected number of observations, then the algorithm attempts to observe the reward of j once more via the selection of the parent of j and parameter N^e is left unchanged [10]. Thus, when the condition specified in step 1 of algorithm 2 is true, this implies that the realizations of the probabilistic feedback graph G are sufficiently good such that the attacker is able to depend on the selected realizations to obtain improved mean estimates accuracy for the various arms. Here, the algorithm is assigned equal probability of either playing an individual arm that satisfies the specified condition in step 1, or to pull the super-arm with equal probability to obtain observations of the various arms.

If the specified condition in step 1 of algorithm 2 is not satisfied by any of the arms in K (i.e. all realizations mean estimates are sufficiently good and highly accurate), then the algorithm proceeds to step 3. In step 3, the algorithm evaluates and determines $C(\cdot)$, which is used to identify the best optimal arm by evaluating the reward gaps between pairs of arms. Thus, if all arms (except the optimal arm) satisfies the condition specified in step 3, then the algorithm is confident that it has identified the optimal arm amongst all the possible arms, and proceeds into step 4 to exploit the optimal arm.

Else, if there is any arm that does not satisfy the conditions specified in step 3, this indicates that there might not be adequate exploration performed for these arms. Hence, the algorithm tries to obtain additional observations on the estimated means for these arms in steps 6 and 7 in order to improve the corresponding mean estimates. As such, when the mean estimates obtained so far are sufficiently close to the true mean values for the arms, then the algorithm proceeds to pull an arm according to the current mean estimates with minimal cost on the

regret as given by the asymptotic lower bound.

Given sufficient time horizon, the algorithm would have performed sufficient exploration rounds to identify the optimal arm amongst all the possible arms, and proceed into step 4 to perform exploitation of the optimal arm. As such, the cumulative regret curve obtained should very likely 'flatten' out as time progresses. During the exploratory phase, the cumulative regret curve would be expected to increase linearly, as a consequence of making pulls of the super-arm and the other sub-optimal arms.

3.3.6 Regret bounds of the LiEtAl algorithm

From [10], the objective of the attacker is to maximize the expected cumulative reward obtained , which is equivalent to minimizing the expected cumulative regret as a consequence of not pulling the optimal arm over time.

The expected cumulative regret function from [10] can thus be defined as follows,

$$R_{\mu}(T;G) = T \max_{i \in V} \theta(\mu_i) - \mathbb{E}\left[\sum_{t=1}^{T} \theta(\mu_{i_t}) \right] \tag{3.7}$$

where the expectation function is over the randomness of r_t and G_t. The mapping from the reward distributions to their respective means is given by $\theta : C \rightarrow \mathbb{R}$. Denote $\theta = (\theta_i : i \in V)$, and assume that there is only one single best optimal action in the set of defined actions, where $\theta_1 > \theta_2 \geq \theta_3 \geq \cdots \geq \theta_K$. Denote $\Delta(\mu) = \Big(\Delta_i(\mu) : i \in V\Big)$, where $\Delta_i(\mu) = \theta_1 - \theta_i$ representing the reward gap between the best optimal action and any other action i. Denote $V^{in}(j) = \{i \in [K] : (i,j) \in E\}$ as the set that represents the incoming neighbors associated with action j.

In [10], the asymptotic lower bound for the algorithm regret is

$$\begin{aligned} C(\mu) = \Bigg\{ c \in [0,\infty)^V : & \sum_{i \in V^{in}(1)} p_{i1} c_i \geq \frac{1}{\mathrm{KL}(\mu_2, \mu_1)} \\ & \sum_{i \in V^{in}(j)} p_{ij} c_i \geq \frac{1}{\mathrm{KL}(\mu_j, \mu_1)}, \quad \forall j \neq 1 \Bigg\} \ . \end{aligned} \tag{3.8}$$

As mentioned in [10], the individual elements contained within the set presented above constitutes the asymptotic pulling "fraction" of arms. This can be used to identify and separate the sub-optimal arms from the best optimal arm.

Hence, the regret bound for the algorithm as presented in [10] is, for any $\epsilon > 0$,

$$\begin{aligned} R_\theta(T) \leq 4\log(T)\sum_{i=1}^{K} c_i(\theta,\epsilon)\Delta_i(\theta) + 10\log(\mathrm{KT}^2)\sum_{i\in V^e}\frac{\Delta_i(\theta)}{p_i^e} \\ +4\sum_{s=0}^{T}\exp\left(-\frac{\beta(s)\epsilon^2}{2K}\right) + 2\beta\left(4\sum_{i=1}^{K} c_i(\theta,\epsilon)\log(T)\right) + 15K, \end{aligned} \tag{3.9}$$

where $c_i(\theta,\epsilon) = \sup\left\{c_i(\theta') : |\theta'_j - \theta_j| \leq \epsilon, \quad \forall j \in [K]\right\}$

It is assumed that $\beta(n) = o(n)$ and $\sum_{s=0}^{\infty}\exp\left(-\frac{\beta(s)\epsilon^2}{2K}\right) < \infty$ for any $\epsilon > 0$. Then, for any θ such that $c(\theta)$ is unique,

$$\limsup_{T\to\infty} R_\theta(T)/\log(T) \leq 4 \inf_{c\in C(\theta)} \langle c, \Delta(\theta)\rangle \tag{3.10}$$

will hold with probability at least $1-\delta$ for any $\delta > 0$.

As observed, the term $O\left(\log(T)\sum_{i=1}^{K}\frac{\Delta_i(\theta)}{p}\right)$ for the expression in the regret bound is due to the gap present between the realizations and the expectations of the probabilistic graphs. Thus, the regret bounds presented are gap-dependent.

3.4 Our Proposed Algorithm

Earlier, we have established that each time the attacker performs a sampling of the super-arm, he might collect intelligence statuses from the respective $\mathcal{K}$ arms although there would be no reward earned. With the collected intelligence, the attacker would then be able to subsequently focus on the more promising arms when the best arms have been identified in order to earn optimal rewards with a high probability in subsequent exploitative actions performed.

While the proposed global confidence bound algorithm in [7] as presented earlier is a possible approach to tackle the aforementioned problem, our numerical analysis noted that the corresponding cumulative regret obtained increases quickly over time. This can be attributed to the logarithmic terms defined in the regret bounds of the global confidence bound algorithm in [7], which causes the cumulative regret to increase rapidly with time. In addition, our numerical analysis on the cumulative regret of the LiEtAl algorithm presented in [10] indicated that the regret increases in constant linear fashion over time when the reward gap between the best two arms is significantly small. This is not optimal for the attacker over time.

Hence, we propose our own algorithm as a novel approach to address the defined problem. The proposed algorithm provides a better strategy for the attacker over time, and the corresponding cumulative regret obtained empirically is consistently lower than those of the other two algorithms. These results are presented in Chapter 4.

3.4.1 Model setting and definition of key parameters

The cumulative regret of the multi-armed bandit algorithms increases in each time step when either a sub-optimal arm or the super-arm is pulled. Not every arm will yield a corresponding observation from the super-arm when it is pulled in practice, since arm k generates samples from a Bernoulli (μ_k) with some probability. Hence, it would not be ideal for the super-arm to be pulled frequently when it not as beneficial, as doing so will result in an increase in the cumulative regret obtained over time. Such an attack strategy would not be optimal for the attacker to exploit the network.

Thus, we propose and incorporate a feasible method for our algorithm to decide on how frequently the super-arm is pulled, depending on how beneficial it is with regards to yielding

observations of the various arms. We aim to achieve an overall cumulative regret that is consistently low over time, by minimizing unnecessary pulls on the super-arm when it is less beneficial.

A definition of the key parameters used in our proposed algorithm is provided next. The notation used may differ from the other algorithms earlier presented.

- η: the probability with which each arm yields an observation from the super-arm.
- α: a parameter used in the algorithm, we set $\alpha = 3/2$ in our work.
- β: a parameter of the algorithm that is used to control how often the super-arm gets pulled.
- K: the total number of arms, $[K] = [1, \ldots, K]$ represents the set of possible arms.
- μ_k: the true mean value of arm k.
- $\mathcal{I}_k^t$: index value of the k arm at the end of round t
- $\mathcal{I}_{sa}^t$: index value of the super-arm at the end of round t
- $\tilde{M}_k^t$: a Binomial random variable corresponding to the number of samples of arm k derived when the super-arm is activated at the end of round t. $\tilde{M}_k^t \sim \text{Binomial}(\nu, N_{sa}^t)$.
- N_k^t: the number of samples of arm k at the end of round t.
- $\tilde{N}_k^t$: the number of individual activations of arm k at the end of round t.
- N_{sa}^t: the total number of activations of the super-arm at the end of round t.
- $\bar{X}_k^t$: the average mean estimate obtained for arm k at the end of round t.
- T: value representing the length of the finite time horizon, where $t \in T$.
- τ: the largest time step value in which the super-arm gets initially sampled.

- ν: a parameter representing the frequency in which the super-arm gets activated.

3.4.2 Model analysis and formulation

We define independent random variables $X_1, \ldots, X_K$ with support on $[0, 1]$, having corresponding mean values $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_K$.

For arm k, let N_k^t be the number of samples and

$$\mathcal{I}_k^t = \bar{X}_k^t + \sqrt{\frac{\alpha \log t}{N_k^t}}, \tag{3.11}$$

be the corresponding arm index by the end of round t. The value of N_k^t increases by 1 each time arm k is individually activated. When the super-arm is activated, N_k^t increases by 1 with probability η.

The index corresponding to the super-arm is

$$\mathcal{I}_{sa}^t = \nu \sum_k \sqrt{\frac{\beta \log t}{N_k^t}}, \tag{3.12}$$

where N_{sa}^t is the number of activations of the super-arm. We assume that $\beta = \alpha$ initially in our analysis, and we seek to determine an optimal value for parameter β as part of our work.

Let $\tilde{N}_k^t$ be the number of individual activations of arm k, so that $N_k^t = \tilde{N}_k^t + \tilde{M}_k^t$, where $\tilde{M}_k^t \sim \text{Binomial}(N_{sa}^t, \eta)$ is a random variable corresponding to the number of samples of arm k derived when the super-arm is activated.

The super-arm gets initially sampled in rounds $t = 1, 2, \ldots, \tau$ while

$$\mathcal{I}_{sa}^t > \max_k \{\mathcal{I}_k^t\}. \tag{3.13}$$

On the events $\left\{|\bar{X}_k^t - \mu_k| < \sqrt{\frac{\alpha \log t}{N_k^t}}\right\}$ and $\left\{|\tilde{M}_k^t - \nu N_{sa}^t| < \eta^t\right\}$ for all arms $k \in [K]$,

$$\tau \geq \tau_{\min} = \min\left\{t \geq 1 : \nu K \sqrt{\frac{\alpha \log t}{\nu t + \eta^t}} \leq \mu_K + 2\sqrt{\frac{\alpha \log t}{\nu t - \eta^t}}\right\} \tag{3.14}$$

since $\mu_K + 2\sqrt{\frac{\alpha \log t}{\nu t - \eta^t}} \geq \mathcal{I}_K^t$ and $\nu K \sqrt{\frac{\alpha \log t}{\nu t + \eta^t}} \leq \mathcal{I}_{sa}^t$.

To further simplify things, we assume that $\eta_t = 0$ for all t's, corresponding to the super-arm deterministically drawing a sample every $1/\nu$ times it is activated. Then we obtain in (3.14),

$$(\nu K - 2)\sqrt{\frac{\alpha \log \tau_{\min}}{\nu \tau_{\min}}} \leq \mu_K. \tag{3.15}$$

No arm gets individually sampled in rounds $t = 1, \ldots, \tau_{\min}$. Arm k does not get individually sampled in any further round if $\mathcal{I}_k^t < \mathcal{I}_K^t$. Therefore, arm k does not get individually sampled in rounds $t = \tau_{\min} + 1, \ldots, \tau_{\max}^k$ where

$$\tau_{\max}^k = \max\left\{t \in \{\tau_{\min} + 1, \ldots, T\} : \mu_k + 2\sqrt{\frac{\alpha \log t}{\nu \tau_{\min}}} < \mu_K\right\}. \tag{3.16}$$

In particular, arms with

$$2\sqrt{\frac{\alpha \log T}{\nu \tau_{\min}}} < \mu_K - \mu_k$$

never get individually sampled.

Replacing for $\tau_{\min}$ in (3.15), we get

$$2\frac{\mu_K}{\nu K - 2}\sqrt{\frac{\log T}{\log \tau_{\min}}} < \mu_K - \mu_k \iff \frac{\mu_k}{\mu_K} < 1 - \frac{2}{\nu K - 2}\sqrt{\frac{\log T}{\log \tau_{\min}}}. \tag{3.17}$$

This expression suggests that the super-arm is beneficial when the number of arms that meet (3.17) is large.

Looking at the expected regret, we have the following additive ingredients:

1. The term induced by $\sum_t P\left(|\bar{X}_k^t - \mu_k| > \sqrt{\frac{\alpha \log t}{N_k^t}}\right)$, which is standard in Upper Confidence Bound proofs.

2. The term corresponding to the super-arm activations, for which we can obtain an upper bound.

3. The term corresponding to individually activating the arms. This can be improved by accounting for the minimum number of super-arm activations.

Thus, a problem that remains for future work is the optimization of the value of parameter β in terms of α in the corresponding expression defined earlier for the super-arm index as part of our work. Another important issue is the optimization of the number of arms that get at least a single sample during the initialization phase; in principle, when the number of arms is very large (say, a million), not every arm needs to get a sample in the initialization stage, because doing so in the presence of Bernoulli sampling for each arm can be inefficient for relatively small values of the parameter ν. How to balance the initialization effort done by the super-arm in terms of the number of arms also is left for future work.

3.4.3 Pseudocode for our proposed algorithm

The pseudocode for our proposed algorithm is presented next.

Algorithm 3: Our Proposed Algorithm

Require: $K,\ \eta,\ \alpha,\ \beta,\ T$

1 Initialize parameters $\alpha,\ \beta,\ \eta,\ t = 0,\ N_k^t,\ N_{sa}^t,\ ,\mathcal{I}_k^t,\ \mathcal{I}_{sa}^t,\ X_k^t$

2 Generate vector of random true mean values μ_k, sorted in decreasing order.

3 Enter initialization phase; sample the super-arm repeatedly until at least a single observation is obtained for every $k-th$ arm. Record the observations. Update parameters N_k^t, X_k^t, and increment t each time the super-arm is sampled.

4 Calculate corresponding index values of the k arms, $\mathcal{I}_k^t = \bar{X}_k^t + \sqrt{\frac{\alpha \log t}{N_k^t}}$.

5 Calculate index value of the super-arm, $\mathcal{I}_{sa}^t = \nu \sum_k \sqrt{\frac{\beta \log t}{N_k^t}}$.

6 Check if the current index value of the super-arm or an individual $k-th$ arm has the largest value overall.

7 If current index value of the super-arm is the largest overall, sample the super-arm repeatedly until every $k-th$ arm has a new single observation obtained. Record the observations. Update parameters N_k^t, X_k^t, and increment t each time the super-arm is sampled. Proceed to step 9.

8 If a $k-th$ arm has the largest overall index value, increment corresponding N_k^t value by 1. Increment t by 1. Update X_k^t and proceed to step 9.

9 Proceed to the next time point $t+1$, if $t < T$.

3.5 Summary

The various algorithms explored as part of this work seek to provide an optimal strategy for the attacker to exploit the network node with the best reward for our defined problem, by minimizing the cumulative regret obtained over the finite time horizon. There exist similarities and differences among the approaches proposed by the various algorithms for the attacker to exploit the best arm identified, in optimal fashion.

The approach adopted by global confidence bound algorithm, as presented in [7], is to perform sufficient initial number of exploration rounds on the various arms within the network in order to identify the promising arms for exploitation. The algorithm continuously per-

forms exploration of the arms until it can identify a single best arm or when sufficient number of exploration rounds have been played [7], before exploiting on the best possible arm. Thus, it "exchanges" the cumulative regret obtained in the initial exploration rounds with reduced error on exploiting the best possible arm over the finite time horizon. As such, the global confidence bound algorithm in [7] achieves an $O\left(T^{\frac{2}{3}} \ln T\right)$ distribution-independent regret. When a unique, optimal arm can be identified, the global confidence bound algorithm achieves $O\left(\log T\right)$ distribution-dependent regret, which is the corresponding theoretical lower bound in classical multi-armed bandit problems [7].

For the LiEtAl algorithm in [10], its approach is to continually assess in every time step, the accuracy of the reward mean estimates obtained for the various arms and also evaluate the corresponding reward gaps, before exploiting on the best arm identified. The LiEtAl algorithm in [10] attempts to identify a best possible arm among the various arms for exploitation. Hence, if the reward gap between the best two arms is small such that the best arm cannot be confidently identified, then the LiEtAl algorithm continues to obtain observations by pulling the arms [10]. This results in the cumulative regret curve increasing linearly over time as it either repeatedly makes pulls on the super-arm or mostly on arms that may possibly be sub-optimal. From [10], the regret bounds are gap-dependent and is of the order $O(\log T)$.

Thus, our proposed algorithm seeks to obtain an improved cumulative regret performance that is consistently lower than those of the other two algorithms for our defined problem. Having a resultant cumulative regret that is not consistently low is not optimal for the attacker, given the finite time horizon. Hence, we adopt the general approach of the global confidence bound algorithm presented in [7], but incorporate a method to determine how frequently the super-arm is pulled. If the number of arms that yield observations from the super-arm is significantly small, which indicates that the super-arm is less beneficial, then our proposed algorithm will make sufficiently fewer pulls on the super-arm during exploration. Our empirical results obtained in Chapter 4 indicates that the cumulative regret performance achieved for our proposed algorithm is consistently lower than those of the other algorithms on our defined problem.

CHAPTER 4:
Results and Analysis

In this chapter, we present the results obtained and analysis for the various multi-armed bandit algorithms explored and discussed in Chapter 3.

We consider the scenario with $K = 100$ arms and a single super-arm. We investigate the cumulative regret performance of the various algorithms on two specific cases with the reward gap for the best two arms being either significantly large or being significantly close. Case 1 has the reward gap being sufficiently small, with the reward difference between the best two arms down to the thousandths place. Case 2 has the reward gap between the best two arms being sufficiently larger, with the reward difference down to the hundredths place.

Table 4.1 provides an example of the reward means for both of these cases.

Table 4.1. True Means for Both Cases.

	Case 1	**Case 2**
Arm #	**p_k (True mean)**	**p_k (True mean)**
1	0.977197714	0.996070817
2	0.972443671	0.975201696

4.1 Global Confidence Bound Learning Algorithm

We evaluate and present the average cumulative regret performance of the global confidence bound algorithm that was obtained over multiple experimental replications via numerical simulation, for our defined scenario with K arms and a single super-arm.

4.1.1 Evaluation of the global confidence bound algorithm

As described by the authors in [7], the global confidence bound algorithm utilizes the following two key conditions to determine whether it should perform exploration of the arms or exploitation in each time step. The first key condition used in the algorithm involves the use of the global confidence bound.

From [7], the expression for the global confidence bound is defined as

$$GCB = \sqrt{\frac{\alpha f_X(t)}{n_\sigma}} \quad , \tag{4.1}$$

where:

α: parameter, $\alpha = 24K$; (K represents the total number of arms)
$\mathcal{X}$: total number of arms and super-arm; $\mathcal{X} = K + 1$
$f_X(t)$: frequency function, $f_X(t) = \ln t + 2 \ln |\mathcal{X}|$
σ: set of actions; $\sigma = \{x_1, x_2, \ldots, x_{|\sigma|}\}$
n_σ: number of exploration rounds played so far

In the algorithm implementation, the global confidence bound is used to evaluate whether there is a significant separation between the estimated means of the optimal arm and the second best arm found so far. If the resulting difference is larger than the calculated value for the global confidence bound so far, then the algorithm is confident on exploiting the current optimal arm in that corresponding time step.

The second key condition used in the global confidence bound algorithm is defined as

$$\left(n_\sigma > t^{\frac{2}{3}} f_X(t) \right)$$

This condition implies that when the total number of exploration rounds played so far is sufficiently large, then the algorithm would also proceed to exploit the optimal arm found so

far. This second condition is used to supplement the first condition as stated earlier. When the difference in sample averages of the optimal arm and second optimal arm found so far is small, which implies implicitly the possibility of multiple near-optimal arms, having only the first key condition would cause the algorithm to constantly enter back into the exploration phase and in turn adds to the cumulative regret obtained over time.

The implementation of the second key condition as an either/or to the first key condition ensures that the algorithm would be able to avoid endless repetitions of the exploration phase the moment that sufficient exploration rounds have been performed, and enter into the exploitation phase.

As presented in [7], when either of the two key conditions are true, the algorithm would be confident to transit from the exploration phase into the exploitation phase, where the optimal arm found so far would be exploited to obtain minimum possible regret.

Parameter η gives the probability with which an observation for an individual arm can be successfully obtained from the super-arm whenever it is pulled. In our analysis, we defined parameter $\eta = 0.05$, which indicates that an observation for an individual arm can be obtained successfully on average five percent of the time when the super-arm is pulled. This is in fact a property of the super-arm in real applications, where critically important information is often times scarce and might not be readily available. In the case of a typical network switch, numerous message data packets from the connected nodes are exchanged with a rapid rate at every time interval and thus requested messages from specific nodes might not be readily available given the busy network load.

The global confidence bound algorithm was also implemented and evaluated for various different values of $K = 10, 100, 1000$ arms. The experimental runs for each value of K was repeated over $N = 100$ independent replications, and the averaged cumulative regret curves over time were plotted as shown in Figure 4.1. The 95% confidence intervals for the cumulative regret curves were evaluated as part of our analysis, but are omitted from Figure 4.1 to keep it uncluttered.

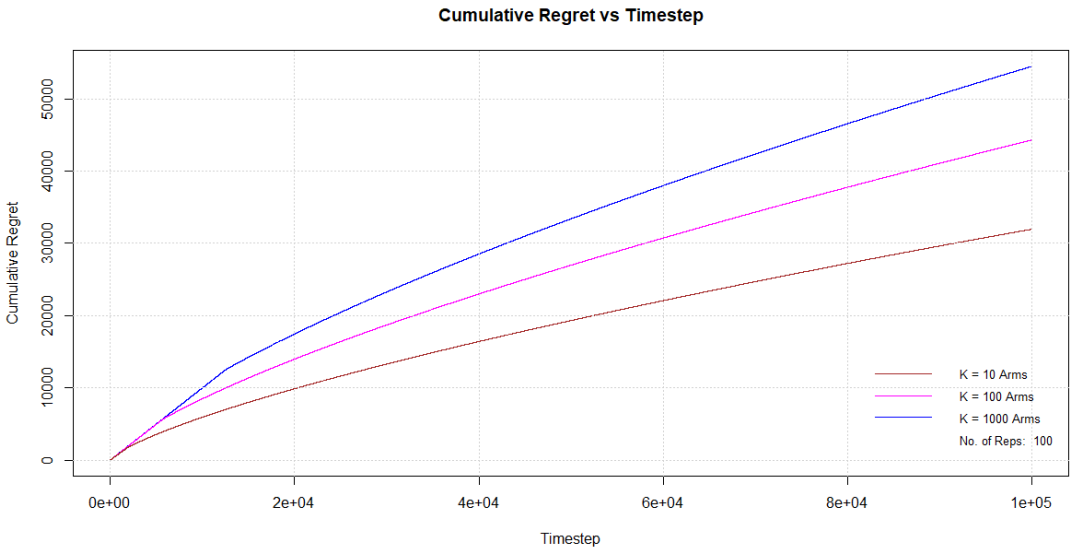

Figure 4.1. Cumulative Regret of the GCB Algorithm versus Time.

As with typical multi-armed bandit algorithms, we observe from Figure 4.1 that the cumulative regret curves exhibit logarithmic growth over time, This is in agreement with the findings presented in [17], where the minimax regret growth for multi-armed bandit problems is of order $O(\log T)$. Thus, the magnitude of the cumulative regret value at time t increases proportionally when the total number of arms K increases.

In the initial time steps, the algorithm is primarily performing exploration of the arms in the network to obtain reliable mean estimates of the associated rewards. This is characterized by the steep growth of cumulative regret exhibited in the various regret curves initially, where more pulls on the super-arm is expected. Over time, the algorithm would have obtained increased number of observations on the probable rewards associated with the respective arms, and therefore would be able to perform exploitation on the most promising arms identified. Given that the Global Confidence Bound (GCB) algorithm was designed to favor more exploration rounds initially to minimize the pulling of sub-optimal arms during exploitation, the cumulative regret curves grow in logarithmic fashion over time. The growth in cumulative regret levels off only when exploitation on the optimal arm is performed.

As such, the GCB algorithm uses the global confidence bound condition to refrain from evaluating reward differences between all possible pairs of arms to improve algorithm

efficiency. In addition, the GCB algorithm performs many more exploration rounds initially in exchange for a reduced probability of error of pulling sub-optimal arms during exploitation phase given that sub-optimal arm pulls would result in an increased regret cost that is linear to the potential number of possible arms.

4.1.2 GCB algorithm regret performance over different reward gap scenarios

We evaluated the cumulative regret performance of the global confidence bound algorithm over two distinct cases for the reward gap difference between the best two arms as mentioned in the beginning of this Chapter.

Figure 4.2 presents the respective averaged cumulative regret curves obtained over 10 experimental repetitions. We chose such small number of replications in order to be able to set a time horizon that is 10 times larger than in Figure 4.1.

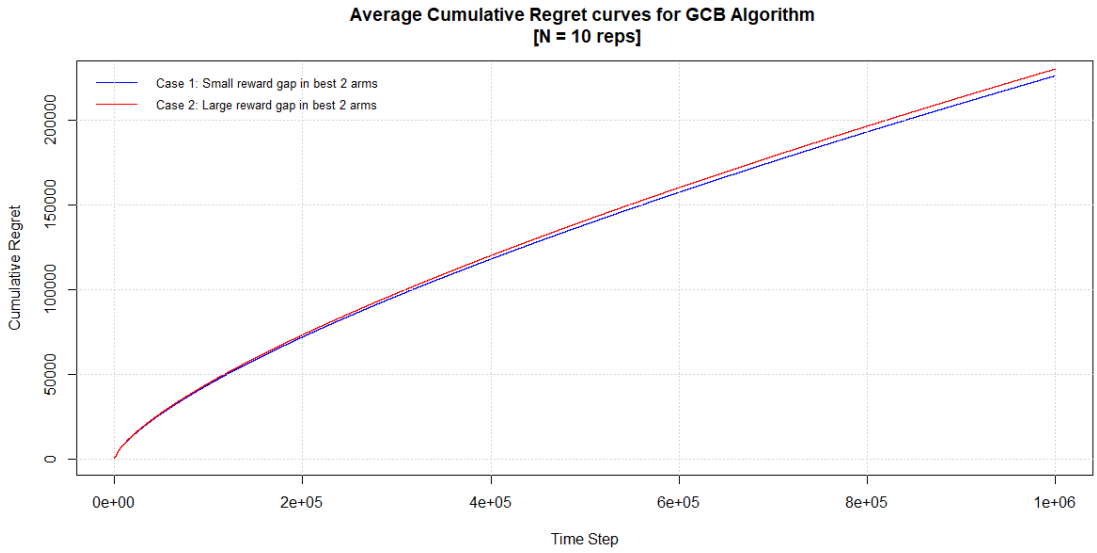

Figure 4.2. GCB Algorithm - Cumulative Regret Curves.

From Figure 4.2, the averaged cumulative regret curves obtained for both cases are relatively close together with small regret value difference over the finite time horizon. This suggests that the global confidence bound algorithm's performance in terms of regret is robust with

regards to the magnitude of the reward gap between the best two arms.

4.2 LiEtAl Algorithm

The LiEtAl algorithm presented by the authors in [10] is more complex in the general setting. Hence, we adapted the algorithm to address our defined problem. We applied the algorithm to our problem with $K = 100$ arms (for computational resource efficiency) and a super-arm to evaluate its corresponding regret bounds obtained. The Uniform distribution was used to generate the corresponding unknown reward distribution of the individual arms for each replication, sorted in descending order such that arm 1 will always provide the maximum expected reward (i.e. zero regret) when pulled.

Correspondingly, for the arms with a larger index, the associated true expected reward values for these arms are small. In addition, whenever the super-arm is sampled, the resultant expected reward is zero and the corresponding regret obtained will be equal to the reward obtained had the best arm been sampled instead. Thus, frequent or unnecessary pulls on these sub-optimal arms and the super-arm is not ideal as doing so significantly contributes to the cumulative regret obtained by the operator over time.

During the initialization process before algorithm execution, the super-arm is sampled repeatedly until there is minimally an observation obtained for each arm. The probability that each arm will yield an observation in each pull of the super-arm is given by parameter η, and we used $\eta = 0.05$ (or 5% of the time) in this case. Hence, it is possible that certain arms might have a better estimate of their true mean values at the end of the initialization due to the multiple samples obtained during this process.

4.2.1 Case 1 - large reward gap between the best two arms

Table 4.2 provides an overview of the key results obtained from the algorithm, for the scenario where the reward gap between the best two arms are is sufficiently large.

Table 4.2. LiEtAl Algorithm - Summary Table 1

Arm #	p_k (True mean)	X_kt (Sample average)	N_kt (no. of times arm was pulled)
1	0.996070817	0.996064425	937415
2	0.975201696	0.971984940	6179
SA	-	-	24198

From table 4.2, we note that the estimated means (sample averages) obtained at the end of the algorithm run are reasonably close to the true mean values for the optimal arm and the second best arm. The algorithm performed the largest number of pulls on the best arm during execution, significantly more than the respective number of pulls on the second best arm and the super-arm. This indicates that the optimal arm was exploited for majority of the time, helping minimize the overall cumulative regret obtained.

Figure 4.3 gives the plot on the selected arm pulled by the algorithm in each time step across the finite predetermined time horizon, for a single sample path.

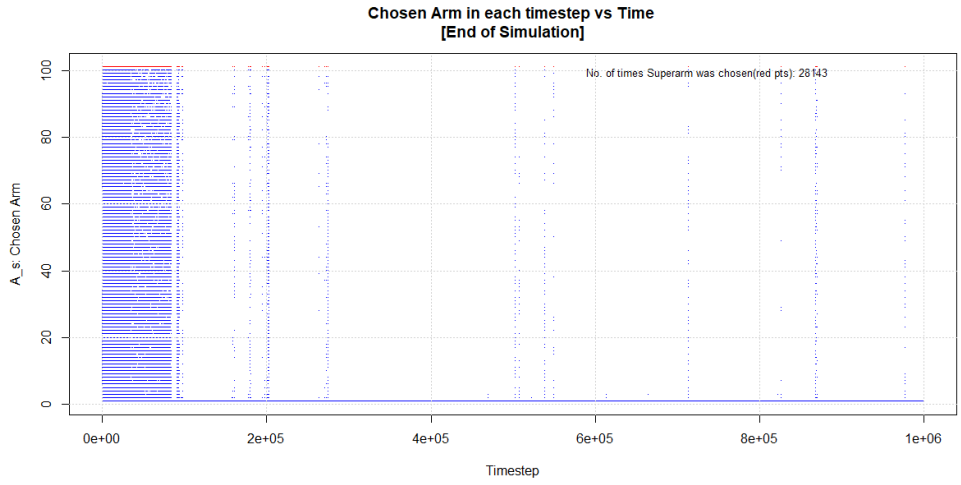

Figure 4.3. Plot of Chosen Arm - Single Sample Path.

The plot shown in Figure 4.3 indicates that the algorithm was making pulls on the super-arm and the respective arms initially to obtain reasonably close mean estimates of the true mean values associated with each arm. Thereafter, the algorithm was performing exploitation on the optimal arm (arm 1, with the largest reward) for most of the time after the exploration phase. It is also observed that the algorithm would also transit back into exploring of the various arms for brief intervals during the exploitation phase. Additionally, pulls on the super-arm by the LiEtAl algorithm mostly occurred in the initial time steps.

Figure 4.4 provides the plot of the average cumulative regret curve obtained versus time.

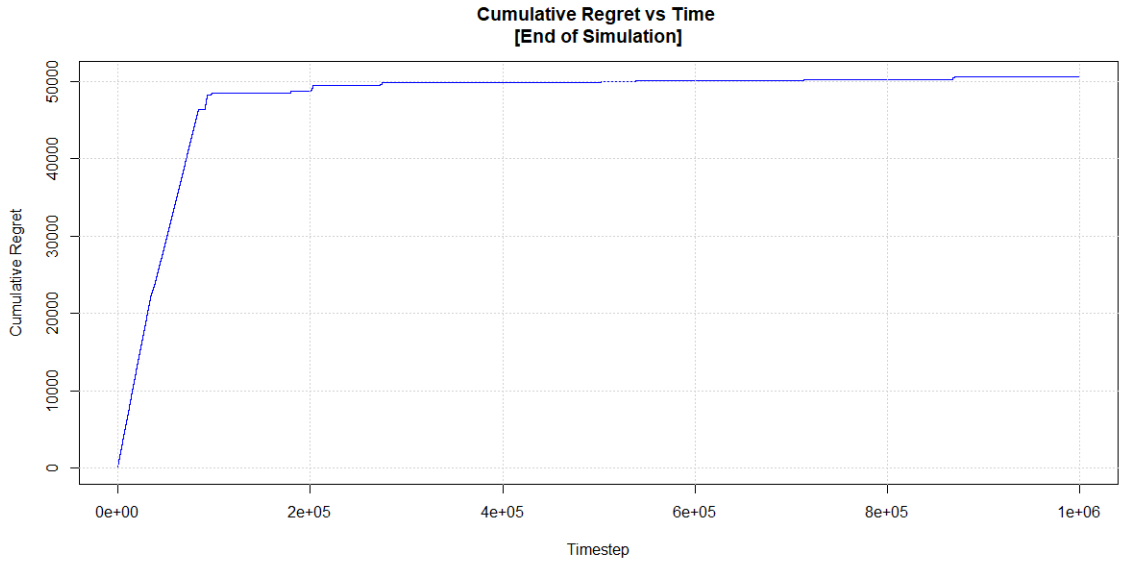

Figure 4.4. Plot of Expected Cumulative Regret versus Time.

From Figure 4.4, it is observed that the cumulative regret obtained increases linearly during the time period when the algorithm was performing exploratory actions. Thus, pulls on the super-arm and the sub-optimal arms during this phase contributes to the linear increment in the cumulative regret obtained. Recall that the regret induced by pulling the super-arm is equivalent to μ_1 (the reward of the best arm), since no reward is earned whenever the super-arm is pulled. When the algorithm is performing exploitative pulls on the super-arm, the cumulative regret curve 'flattens' out as shown in the plot.

4.2.2 Case 2 - small reward gap between the best two arms

Earlier, we have demonstrated that the LiEtAl algorithm performs reasonably well when the true reward means for the best two arms are sufficiently far apart. However, if the unknown reward distribution results in the reward gap between the best two arms to be sufficiently small, our analysis indicates that the cumulative regret achieved by the execution of the algorithm grows 'linearly' and is not optimal to the operator. We use $K = 100$ arms in our analysis for computational efficiency reasons.

Table 4.3 provides an overview of the key results obtained for this particular analysis.

Table 4.3. LiEtAl Algorithm - Summary Table 2

Arm #	p_k (True mean)	X_kt (Sample average)	N_kt (no. of times arm was pulled)
1	0.977197714	0.976682993	187851
2	0.972443671	0.972309300	96312
SA	-	-	330019

From Table 4.3, the true reward mean values associated with optimal arm (arm 1) and the next best arm (arm 2) are observed to be significantly close together, differing in the thousandths place. While the optimal arm had almost double the number of pulls by the algorithm in comparison with the next best arm, we observed that the algorithm was instead pulling the super-arm most of the time. The number of pulls on the respective arms serves as a preliminary indication that the algorithm was unable to clearly distinguish the optimal arm from the next best arm, given the significantly small reward gap. It is likely that the algorithm was attempting to utilize the super-arm to obtain additional observations on the respective arms, before determining that arm 1 was indeed the optimal arm for exploitation.

Figure 4.5 presents the plot on the selected arm pulled by the algorithm in each time step across the defined time horizon, for a single sample path. As observed, the algorithm was continuously performing exploration, by pulling either the super-arm or the individual arms.

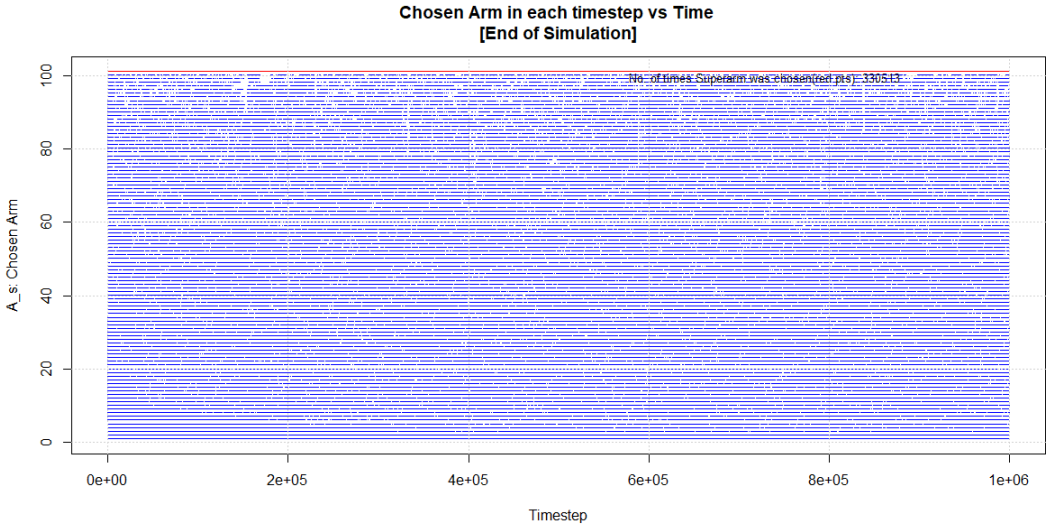

Figure 4.5. Plot of Chosen Arm versus Time - Single Sample Path.

Figure 4.6 shows the plot of the average cumulative regret curve obtained versus time at the completion of the algorithm run.

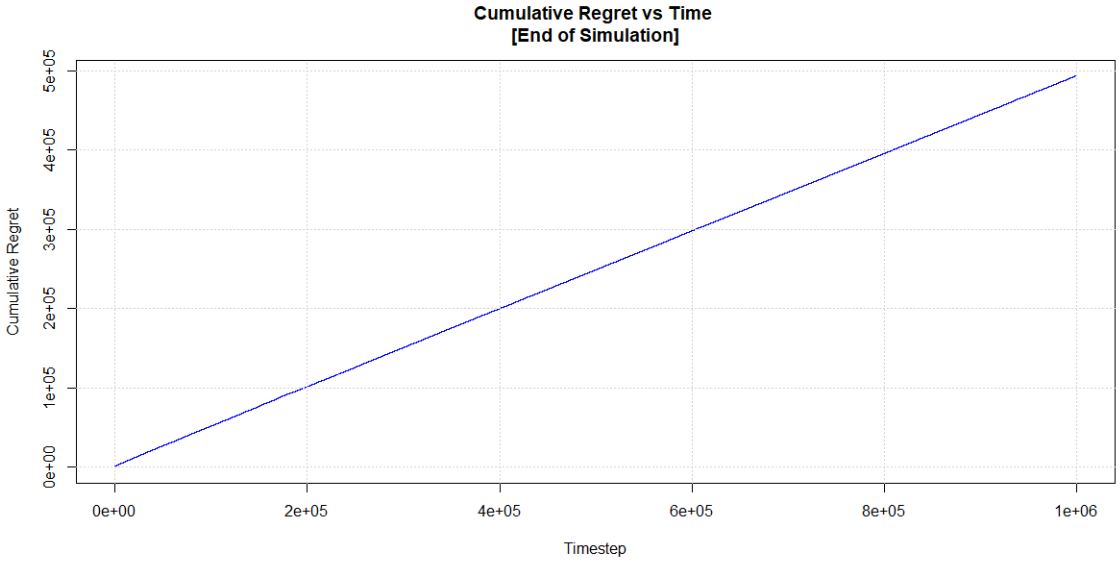

Figure 4.6. Plot of Expected Cumulative Regret versus Time.

We observe that the resultant cumulative regret curve obtained increases linearly over time, which further indicates that there were many pulls of the sub-optimal arms or that the super-arm was used majority of the time. Hence, in the case when the reward gap between the best two arms is significantly small, the algorithm did not succeed in clearly distinguishing the optimal arm over the other sub-optimal arms. As such, it did not manage to continually perform exploitation on the optimal arm over time.

4.2.3 LiEtAl algorithm regret performance over different reward gaps

In the preceding sub-sections, we presented the results and analysis on the cumulative regret obtained by the LiEtAl algorithm depending on the reward gap of the best two arms. Our analysis noted that when the reward gap between the best two arms is significantly small down to the thousandths place, the cumulative regret curve increases linearly over time.

Figure 4.7 shows the plot of cumulative regret curves obtained for the different reward gap scenarios explored with the LiEtAl algorithm. There was decisive exploitation on the optimal arm when the reward gap was large, resulting in the significant difference in cumulative regret obtained for both cases.

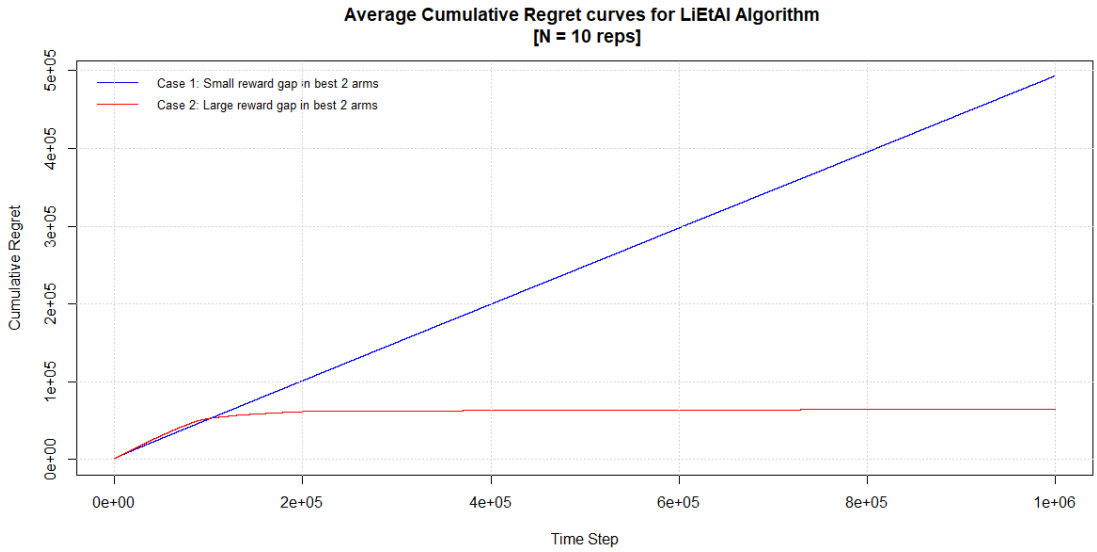

Figure 4.7. LiEtAl Algorithm - Cumulative Regret Curves.

4.2.4 Parameter (β) space exploration for the LiEtAl algorithm

From the results presented in the earlier sub-section, we find that the LiEtAl algorithm had a cumulative regret curve that was observed to be increasing linearly when the difference in true reward means associated with the optimal arm and the next best optimal arm was sufficiently small. For the operator, this would not be ideal as it would mean that the operator did not successfully identify the optimal arm amongst the various possible arms during the exploration phase and hence made pulls on the other sub-optimal arms during exploitation phase. Hence, the operator would not be able to maximize his expected reward obtained over time.

As such, we made possible explorations on the auxiliary function value parameter β that is utilized in step 5 of algorithm 2. The condition specified in this particular step of the algorithm serves as a condition to determine whether an individual arm has adequate number of mean observations obtained over time through sufficient exploration rounds. Thus, if the number of observations obtained for an arm is deemed to be insufficient, then the algorithm will attempt to obtain an additional observation for the arm via a new exploration round and increases the count of parameter N^e. Hence, the auxiliary function parameter $\beta : \mathcal{N} \rightarrow [0, \infty)$ is used to steer the exploration performed by the algorithm in order to have the mean estimates to be reasonably close to the true means.

Our objective was to attempt to vary the value of parameter β to get the mean estimates of the arms to be as close as possible to the associated true mean values. By doing so, we attempt to investigate whether this assists the algorithm to clearly distinguish the optimal arm from the other less promising arms in order to enter the exploitation phase quickly to minimize the overall cumulative regret. We used the same reward distribution from our earlier analysis, with the true means of the best two arms being significantly close together, for our exploration into the β parameter space and investigate the corresponding cumulative regret obtained.

Figure 4.8 shows the plot of the respective cumulative regret curves obtained for different values of parameter β that was explored.

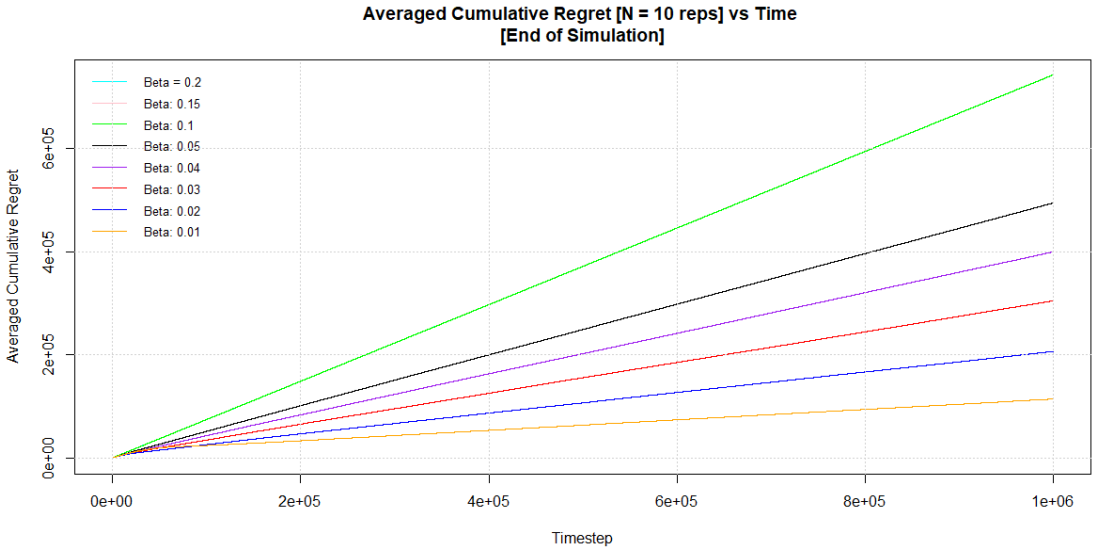

Figure 4.8. Plot 1 of Cumulative Regret Curves for Various β's.

From Figure 4.8, when the value of β is increased, the value of the cumulative regret obtained over time increases linearly, and is not optimal for the operator. Thus, the algorithm would need to perform additional exploration rounds to obtain increasingly accurate estimates of the true reward means, thus adding to the cumulative regret over time.

Another observation made with reference to the plot in Figure 4.8 pertains to the explorations performed when the value of parameter β was reduced. We find that the cumulative regret obtained over the predetermined time horizon decreases when the value of parameter β was correspondingly reduced. However, the cumulative regret curves obtained for the smaller values of β explored also exhibit a linearly increasing trend in regret over time.

Therefore, we next focus our analysis on investigating the cumulative regret performance obtained for the algorithm for smaller values of parameter β. Figure 4.9 shows the plot of the various cumulative regret curves obtained for even smaller values of parameter β explored.

From Figure 4.9, we observe that for certain small values of parameter $\beta \leq 0.004$, the cumulative regret curves obtained was 'flattening' out after exhibiting a linear increase in regret for the initial time duration. Similar to what we would expect from the performance of this algorithm, there was an exploration of various arms being performed initially before executing the exploitation of the optimal arm as shown in the regret curve.

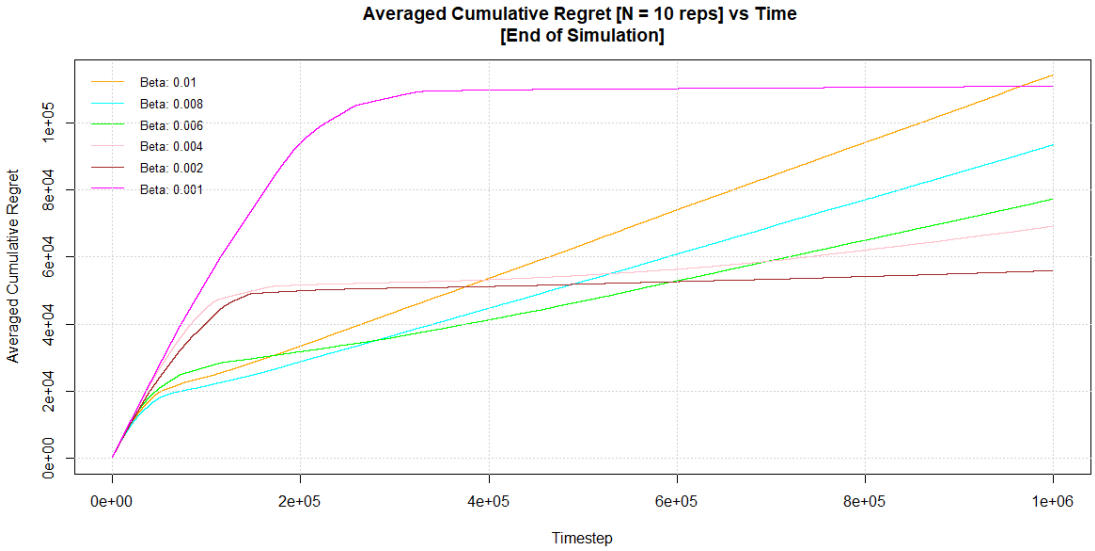

Figure 4.9. Plot 2 of Cumulative Regret Curves for Various β's.

Hence, a preliminary observation obtained from the plot is that the significant reduction in the value of parameter β serves as a 'relaxation' of the criterion used by the algorithm to determine the accuracy required for the estimated mean values of arms. This in turn enables the algorithm to enter the exploitation phase much faster to perform pulls on the optimal arm identified to minimize the cumulative regret obtained, even for the case when the true means of the best two arms are significantly close.

However, we also observe that for the case when $\beta = 0.001$, the cumulative regret curve exhibits the same 'shape' of performing both exploration of arms and exploitation of the optimal arm. But, the cumulative regret value obtained is much larger than for the other values of β explored. This suggests that the algorithm was largely making pulls on the super-arm as part of its exploration in the initial time steps.

4.2.5 Exploration of the constant value used in reward gap evaluation

In step 3 of algorithm 2, we utilized an expression to determine the set of "fraction" of arms that have a significantly identifiable reward gap, and this is used as a metric for the algorithm to distinguish the arm with the optimal reward amongst all possible arms. This was theoretically defined as the asymptotic lower bound for the algorithm as presented earlier in (3.8).

The expression for $C_k(t)$ is given as follows,

$$C_k(t) = \frac{N_k(t)}{16\log(t-1)} \quad , \tag{4.2}$$

where $N_k(t)$ gives the total number of times an arm k was pulled so far up till time step t.

In [10], we note that the constant used in the denominator of (4.2) was of value 16. From the analytical results and various plots presented earlier, we note that in the case where the true means of the best two arms were significantly close with negligible difference in the thousandths place, the algorithm did not manage to identify the optimal arm confidently. It either failed to identify arm 1 as the optimal arm, or it did not make a significant distinction between arms 1 and 2 on which was apparently the optimal arm. As such, our earlier results demonstrated that the algorithm was somewhat not sufficiently robust in minimizing cumulative regret obtained over time in our application.

Hence, in our exploration of the parameter space of the algorithm, we attempted variations in the constant value defined in the denominator of the expression given in (4.2), and investigated the effect on the cumulative regret curves obtained for the same reward distribution in which the difference in the true means of the best two arms was significantly negligible. We present our results in the following plots as part of our analysis and discussion.

Figure 4.10 shows the plot of the averaged cumulative regret obtained over 10 experiment replications, for a subset of constant values explored.

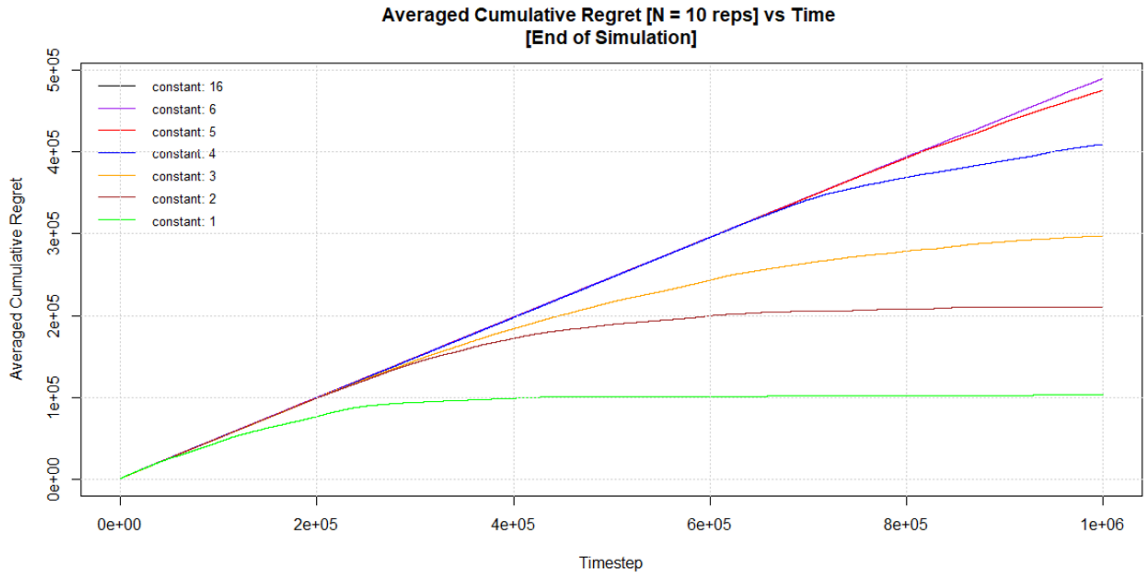

Figure 4.10. Varying Constant Value: Plot of Cumulative Regret versus Time.

From our analysis, we note that when the constant value in the denominator of the expression in (4.2) is increased, the averaged cumulative regret was observed to be increasing linearly over time and hence not beneficial to the operator in maximizing potential expected reward. In addition, we explored decreasing the value of the constant, and investigated the expected cumulative regret obtained.

We note that when the value of the constant is decreased slightly, there is no significant improvement in the expected cumulative regret obtained over time. However, when the constant value is significantly decreased, we see from the plot presented in Figure 4.10 that the expected cumulative regret curve starts to show signs of 'flattening' out. As the value of the constant was decreased further, the resultant cumulative regret curve 'flattens' out even more significantly.

This suggests that the algorithm did manage to identify the most promising arm in this case and exploited it over time. We note that the smaller the value of the constant used in the denominator of (4.2), the smaller the expected cumulative regret value obtained over time. Thus, when the constant value is reduced, it serves as a relaxation on the condition used by the algorithm to evaluate the reward gaps between the sub-optimal arms and the optimal arm.

4.3 Our Proposed Algorithm

Earlier, we have established that each time when the operator performs a sampling of the super-arm, he would be able to collect intelligence statuses for the respective K arms although there would be no expected reward earned. With the collected intelligence, the operator would then be able to subsequently focus on the more promising arms when the best arms have been identified in order to earn optimal rewards with a high probability in subsequent exploitative actions performed.

While the proposed GCB algorithm in [7] as illustrated in Section 4.1 is a possible approach to tackle the aforementioned problem, our analysis noted that it is not as efficient in minimizing the cumulative regret obtained compared to the other algorithms. We attempt to develop a more efficient approach to address the problem presented in our defined scenario.

4.3.1 Proposed Algorithm Regret Performance over different reward gap scenarios

Figure 4.11 presents the cumulative regret curves for the cases where the reward gap between the best two arms either significantly large or small, as per similar analysis for the other algorithms.

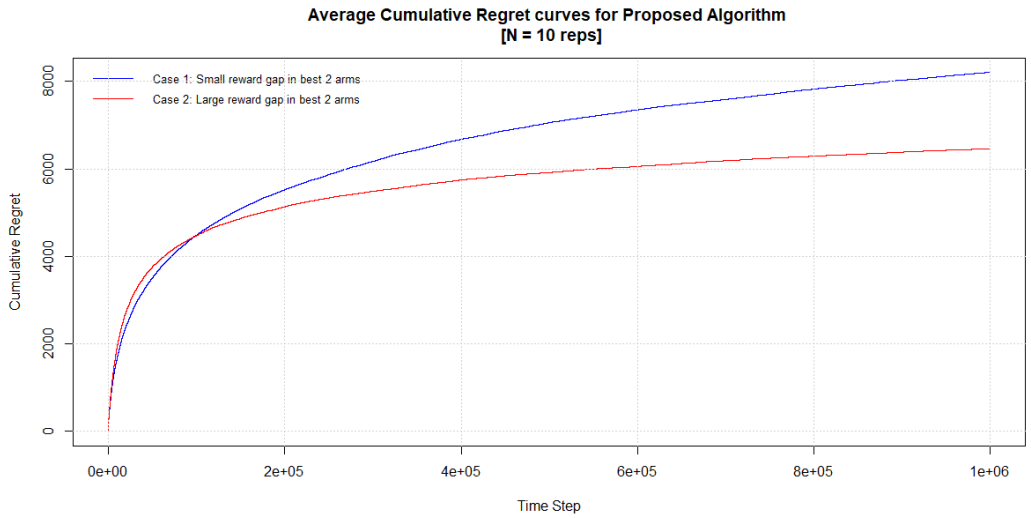

Figure 4.11. Proposed Algorithm - Cumulative Regret Curves.

From Figure 4.11, we observe that the cumulative regret obtained over time for the reward means in significantly reduced, in comparison to the regret presented for the GCB algorithm in Figure 4.2. The case in which the reward gap between the best two arms was significantly larger had a worse overall performance, with increased cumulative regret obtained over time in comparison with the other case analyzed.

4.3.2 Determining possible optimal value range for parameter β

In the evaluation of the GCB algorithm, we mentioned that the cumulative regret bound for our problem with K arms and a super-arm should be at least as good as the regret bound of that obtained by the Upper Confidence Bound 1 (UCB1) algorithm. In addition, we have shown that the number of times the super-arm is used in t rounds, S_t, is dependent on α, β and η. Assuming that the parameters η and α are fixed, an optimal value of β has to be derived in order to ensure that we can achieve corresponding cumulative regret to be of order $O(\log(T))$. Thus, β is a parameter in the algorithm that is used to control how often the super-arm gets when the algorithm is in the exploration phase.

To evaluate and determine a possible optimal range of values for parameter β, we let parameter $\alpha = 3/2$ in our analysis. There are $K = 100$ arms and 1 super-arm. For various selected values of parameter β in our proposed algorithm, we obtained the expected cumulative regret bounds over 10 experimental replications for values of $\eta = [0, 1]$. The various cumulative regret bounds are as shown in Figure 4.12 below.

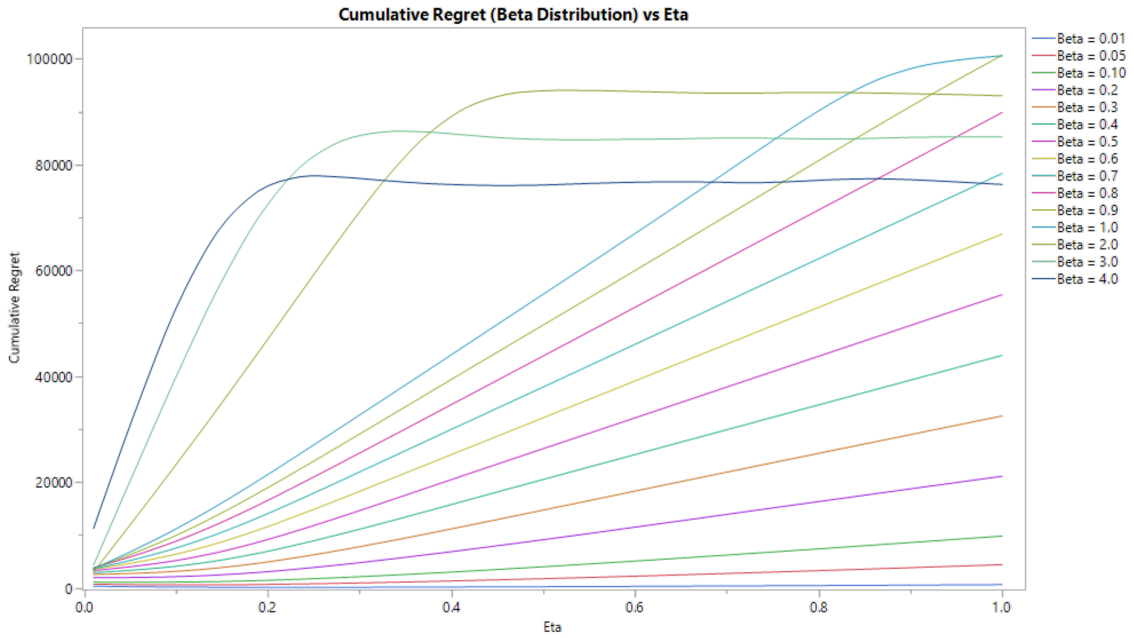

Figure 4.12. Cumulative Regret Curves (with Super-arm) for Various β versus η.

The plot of the respective expected cumulative regret bounds for various values of β versus η in Figure 4.12 seems to suggest that in order to minimize the expected cumulative regret, it would be ideal to keep the value of parameter β as small as possible.

A heatmap of the expected cumulative regret values was generated for the parameter values of $\eta = [0, 1]$ and $\beta = [0, 1]$ to determine the values region of interest in which the expected cumulative regret will be kept small. From the heatmap, it can be observed that the expected cumulative regret will be the minimum when the values of η and β are small. The generated heatmap of the expected cumulative regret values is as shown in Figure 4.13.

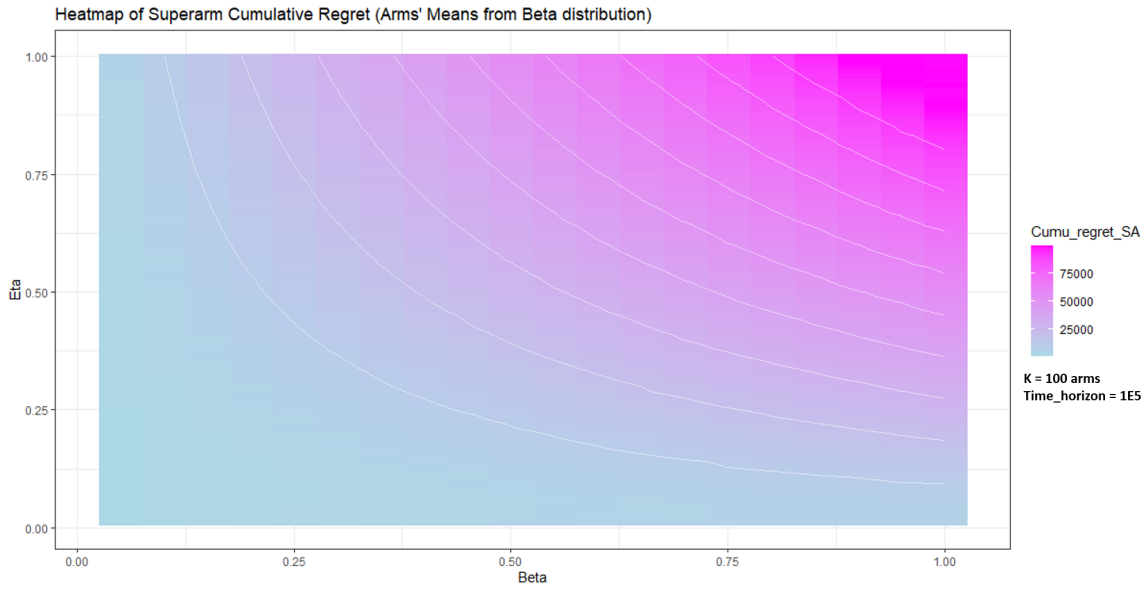

Figure 4.13. Cumulative Regret Heatmap for Parameter Values of η versus β .

Figure 4.13 provides a quick overview on the optimal value choices for parameter β for fixed values of η. It would be optimal for the chosen β value to be small to achieve minimal cumulative regret if η is large. Hence, as part of the analysis, we considered a range of parameter values for $\beta = [0.001, 0.5]$ and $\eta = [0.01, 0.1]$ in order to optimize the selection of possible value range for β so as to keep the expected cumulative regret bound to a minimum.

Figure 4.14 shows the plot of $\beta(\eta)$ versus η parameter values, which was also utilized in the process for optimizing the selection of algorithm parameters β and η. $\beta(\eta)$ was obtained as the β value that corresponded to the minimum expected cumulative regret value derived over $N = 1000$ experimental replications for the individual values for the range of $\eta = [0.01, 0.1]$.

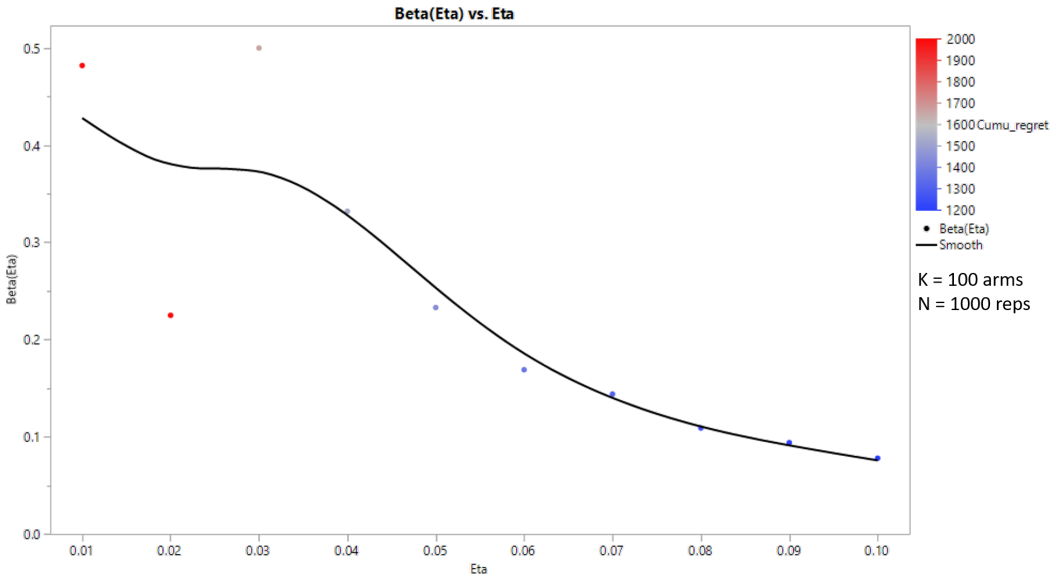

Figure 4.14. Plot of $\beta(\eta)$ versus η.

From Figure 4.14, it can be observed that when the value of environment parameter η increases, the corresponding choice of value for parameter β should decrease in order to achieve minimum expected cumulative regret. This is in tandem with the insight obtained earlier for the optimal choice for the range of values for parameters β and η as presented in Figure 4.13.

Further analysis was performed for the expected cumulative regret bounds of the respective η values for $K = 100$ arms and 1 super-arm, in comparison to the corresponding expected cumulative regret bound for the case without the super-arm. In the case without the super-arm, the expected cumulative regret bound obtained would be as good as the regret obtained by the UCB1 algorithm for multi-armed bandit models.

Our research indicates that the cumulative regret performance of the proposed algorithm is associated with the value of parameter β. In practice, the performance of the proposed algorithm may improve by varying the value of parameter β to minimize cumulative regret obtained. However, we find that the suggested degree to which parameter β is varied is dependent on the value of η.

Figure 4.15 shows the plot of the respective expected cumulative regret bounds for $\eta = 0.01, 0.02, 0.03$ and for the case without the super-arm.

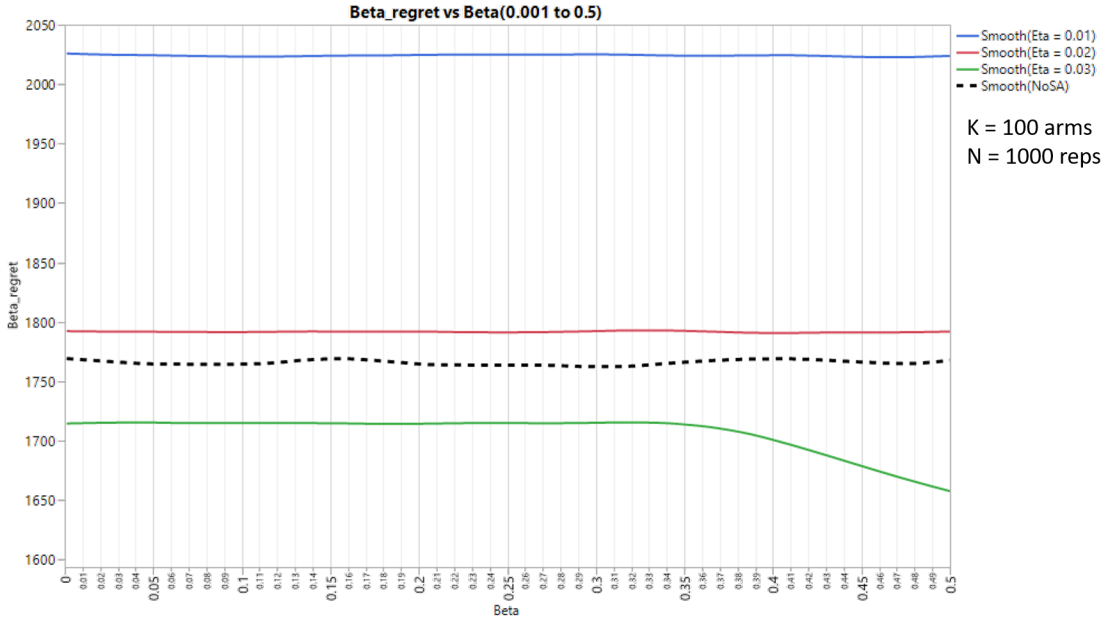

Figure 4.15. Selected Cumulative Regret Curves for η without Super-arm.

Clearly, it is observed from the plot that for $\eta = 0.01, 0.02$, the expected cumulative regret obtained is worse than the case without the super-arm across the range of β values. For $\eta = 0.03$, the corresponding expected cumulative regret bound obtained is lower than that for the case with the super-arm.

The expected cumulative regret bounds for $\eta = [0.01, 0.1]$ and for the case without the super-arm (equivalent to the regret bound obtained for the UCB1 algorithm) across the range of $\beta = [0.001, 0.5]$ is as shown in Figure 4.16.

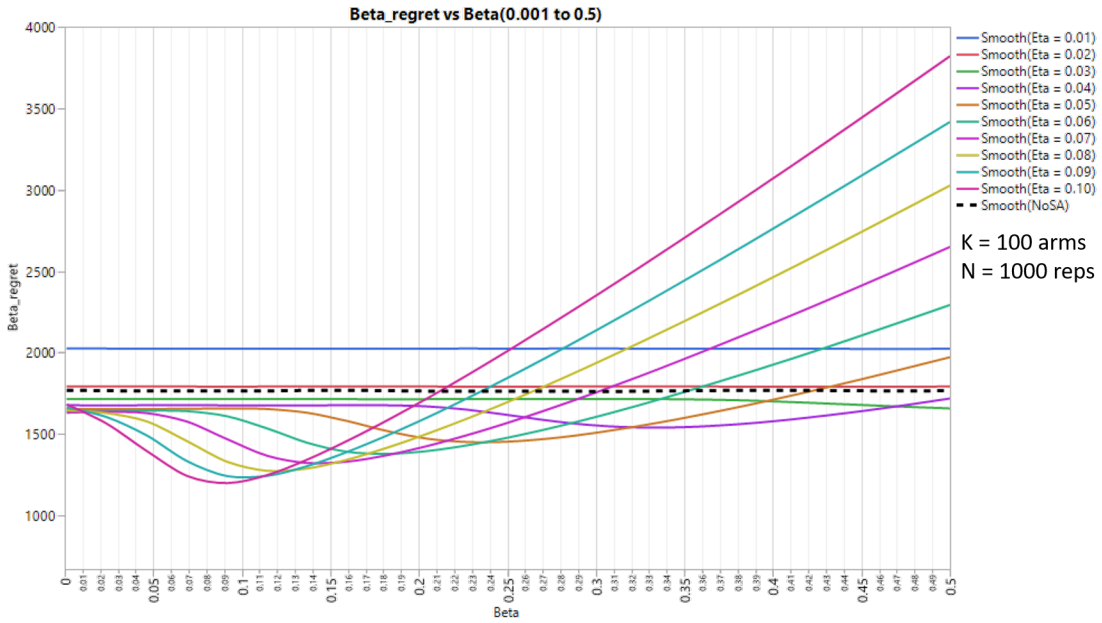

Figure 4.16. Cumulative Regret Bounds for Various η, without Super-arm.

From Figure 4.16, for increasing values of η it is observed that the minima of the respective cumulative regret curves correspond to decreasing value choices of parameter β. Hence, we find that the algorithm achieves smaller cumulative regret when β is optimally varied in accordance with the corresponding value for η. We thus conclude that when η is large, the super-arm should be pulled by the operator less frequently (as given by the optimal values for β to be correspondingly small) in order to minimize the overall cumulative regret obtained.

4.3.3 Cumulative Regret comparison across all algorithms explored

We proceed to compare and evaluate the performances of the various algorithms explored, in terms of the cumulative regret, in application to our earlier defined scenario.

The plot of the respective average cumulative regret curves obtained over 100 different random reward uniform distributions, is as presented in Figure 4.17.

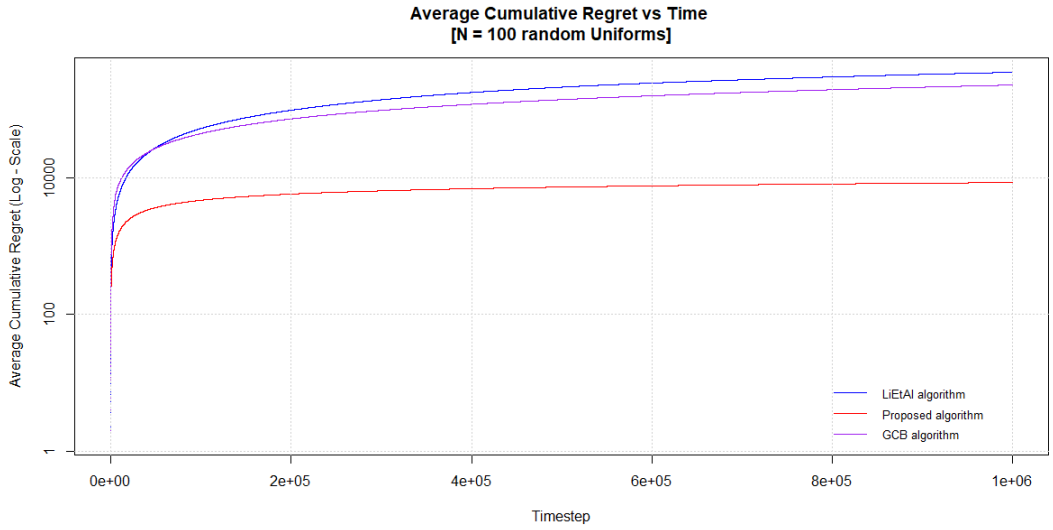

Figure 4.17. Averaged Cumulative Regret Curves for Various Algorithms.

The hundred randomly generated uniforms of the random rewards distributions for the arms, are used to provide an objective comparison of the cumulative regret performance of the various algorithms explored in this work. These random uniform distributions comprise of a mix of distributions with means for the best two arms being either significantly close together, as well as the means for the best two arms being significantly far apart.

We observe from Figure 4.17 that the LiEtAl algorithm had the largest cumulative regret obtained over time. On the other hand, our proposed algorithm achieved the best performance amongst the algorithms explored in terms of regret. Thus, this indicates that our proposed algorithm potentially serves as a good framework for formulating optimal network attack strategies where minimum cumulative regret can be achieved.

CHAPTER 5:
Conclusion

In this chapter, we present a summary of the key results obtained in this work. These results were derived analytically as part of Chapter 3 and numerically verified in Chapter 4 of this work.

We consider a typical LAN topology with a single network switch (depicting the super-arm) and K connected network nodes (depicting the set of base arms). We propose a stylized model for an attacker to perform strategic exploitation of the network without prior knowledge of its topology, via the application of relevant multi-armed bandit models. The proposed model seeks to optimize the attacker's sequence of arm pulls, in order to minimize the total cumulative regret obtained over a finite time horizon.

In our defined scenario, the reward of each individual arm is random from respective Bernoulli distribution with unknown mean. The attacker has no knowledge of the reward distributions and the corresponding means of the arms' rewards. However, the attacker has the choice of pulling either the super-arm or any of the individual arms in each time step.

When the attacker pulls an individual arm, he accrues a regret equivalent to the difference of the best possible reward and the current reward obtained. If the attacker decides to pull the super-arm, he earns a regret corresponding to the best possible reward. Pulling the super-arm results in a Bernoulli sampling from a random set of arms, with the observations obtained providing possible intelligence on these arms. Not every arm will yield an observation when the super-arm is pulled.

We evaluate the cumulative regret performance of two algorithms, namely the Global Confidence Bound algorithm and the LiEtAl algorithm. We consider two main scenarios, defined when the reward gap between the best two arms are either significantly large or small. This translates to whether a single, unique best arm can be distinctly identified for exploitation by the algorithm to minimize overall cumulative regret.

In general, these multi-armed bandit algorithms seek to perform adequate initial exploration rounds to achieve reduced probability of error during exploitation of the best arm. This serves

as a trade off to the cumulative regret obtained during exploration in the initial time steps. Additionally, the algorithms also attempt to identify the best arm based on the reward gaps between pairs of arms, and only exploit the best arm when it is clearly distinguishable amongst the arms.

We also developed a novel algorithm that seeks to improve upon the cumulative regret performance of the Global Confidence Bound algorithm. We devised and incorporated a method to decide on the frequency of pulling the super-arm, depending on how beneficial the super-arm is in relation to yielding observations from the individual arms. Our research work demonstrates that the proposed algorithm achieved significantly lower cumulative regret over time in comparison with the other multi-armed bandit algorithms evaluated.

Our stylized model serves as a promising approach that is suitable to be adopted as a framework for devising optimal network exploitation strategies that achieve consistently small cumulative regret over a finite time horizon. Parts of the theoretical analysis for the proposed algorithm are left as part of future work.

List of References

[1] L.Brent, "NATO's role in cyberspace," NATO Review, February 12, 2019 [Online]. Available: https://www.nato.int/docu/review/articles/2019/02/12/natos-role-in-cyberspace/index.html

[2] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000014000178

[3] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, December 1933 [Online]. Available: https://doi.org/10.1093/biomet/25.3-4.285

[4] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, September 1952 [Online]. Available: https://projecteuclid.org/journals/bulletin-of-the-american-mathematical-society-new-series/volume-58/issue-5/Some-aspects-of-the-sequential-design-of-experiments/bams/1183517370.full

[5] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020.

[6] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[7] T. Lin, B. Abrahao, R. Kleinberg, J. Lui, and W. Chen, "Combinatorial partial monitoring game with linear feedback and its applications," in *Proceedings of the 31st International Conference on Machine Learning*, 2014 [Online]. Available: http://proceedings.mlr.press/v32/lind14.html

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002 [Online]. Available: https://doi.org/10.1023/A:1013689704352

[9] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *Proceedings of the 30th International Conference on Machine Learning*, 2013 [Online]. Available: http://proceedings.mlr.press/v28/chen13a.html

[10] S. Li, W. Chen, Z. Wen, and K.-S. Leung, "Stochastic online learning with probabilistic graph feedback," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, February 2020 [Online]. Available: https://www.microsoft.com/en-us/research/publication/stochastic-online-learning-with-probabilistic-graph-feedback/

[11] R. Elderman, L. J. J. Pater, A. S. Thie, M. M. Drugan, and M. Wiering, "Adversarial reinforcement learning in a cyber security simulation," in *ICAART 2017 - Proceedings of the 9th International Conference on Agents and Artificial Intelligence*, January 2017 [Online]. Available: https://www.ai.rug.nl/~mwiering/GROUP/ARTICLES/CyberSec_ICAART.pdf

[12] Y. Qian, C. Zhang, B. Krishnamachari, and M. Tambe, "Restless poachers: Handling exploration-exploitation tradeoffs in security domains," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2016 [Online]. Available: https://www.cais.usc.edu/wp-content/uploads/2017/05/p123.pdf

[13] K. Liu and Q. Zhao, "Dynamic intrusion detection in resource-constrained cyber networks," in *2012 IEEE International Symposium on Information Theory Proceedings*, July 2012 [Online]. Available: https://ieeexplore.ieee.org/document/6284708

[14] T. Jung, S. Martin, D. Ernst, and G. Leduc, "Contextual multi-armed bandits for web server defense," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, June 2012 [Online]. Available: https://ieeexplore.ieee.org/document/6252760

[15] M. Gutierrez, J. Cerný, N. Ben-Asher, E. Aharonov-Majar, B. Bosanský, C. Kiekintveld, and C. Gonzalez, "Evaluating models of human behavior in an adversarial multi-armed bandit problem," in *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation*, July 2019 [Online]. Available: https://mindmodeling.org/cogsci2019/papers/0088/index.html

[16] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems*, December 2011 [Online]. Available: https://proceedings.neurips.cc/paper/2011/hash/e1e32e235eee1f970470a3a6658dfdd5-Abstract.html

[17] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, March 1985 [Online]. Available: https://www.sciencedirect.com/science/article/pii/0196885885900028

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California